UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Reconstrucción 3D del cuerpo humano mediante puntos de referencia

TESIS

Que para obtener el título de

Ingeniero en Computación

P R E S E N T A

Mauricio Eduardo Negrete Rodríguez

DIRECTOR(A) DE TESIS

Dr. Miguel Ángel Padilla Castañeda

Ciudad Universitaria, Cd. Mx., 2018

Dedicatoria

Con todo corazón a las tres personas que siempre han estado a mi lado, quienes formaron la persona que soy hoy en día, siempre enseñándome nuevas cosas, mostrándome lo más posible de este mundo y sobre todo brindándome amor y apoyo en todo lo posible: mi mamá Magdalena, mi abuelita Socorro y mi tía Ethel.

Agradecimientos

A mi hermano Jorge Iván, que me ha brindado todo su apoyo en todo momento.

Sin duda alguna, a la familia de mi madre que siempre me ha mostrado su afecto.

A mi gran amigo y hermano de alma: Armando, que me ha enseñado lecciones importantes para mi vida, compartiendo grandes recuerdos a través de los años.

A mi amiga Laura que durante el curso carrera fue un gran apoyo de muchas maneras, proporcionándome su confianza e invaluable amistad.

A todos mis amigos y amigas del CCH SUR, Servicio Social, Facultad, ICAT y en la UNAM con quienes tengo y espero tener más increíbles recuerdos.

A mis amigas Alexa y Diana que me apoyaron en el proceso de esta tesis y ofreciéndome su gran amistad en tan poco tiempo.

Al Dr. Miguel Ángel Padilla Castañeda y al Dr. Alfonso Gastélum Strozzi por su guía en el desarrollo de esta tesis.

Al licenciado en ingeniería Iván Uriel que me permitió convivir con él durante la carrera.

Agradecimiento al Proyecto PAPIME-PE109018 por la beca recibida y beca interna ICAT-UNAM.

Contenido

Tabla de Figuras

1 Introducción

En la actualidad para generar un modelo 3D de un objeto, persona o escena que se desea, existen diversos tipos de tecnologías (máquina de medición por coordenadas, cámaras basados en láseres, entre otros) con la finalidad de adquirir los datos necesarios.

Los modelos 3D llegan a tener una gran variedad de aplicaciones, desde video juegos hasta simuladores y aplicaciones en medicina, pero llega a ser muy costoso el equipo que se emplea llegando a tener precios que varían entre los $250 - $30,000 dólares[1][2][3], por lo tanto, el sensor Kinect v2 es una opción más accesible al ofrecer características que permiten generar modelos 3D a un bajo costo.

El Kinect[4] es la serie de cámaras producidas por Microsoft para la detección de movimiento para su consola de video juegos Xbox y Xbox One, cuenta con una cámara RGB y un sensor de profundidad que permiten realizar la reconstrucción 3D. La segunda versión (Kinect v2) ofrece mayor precisión, un mayor campo de visión, imágenes de color a alta calidad, entre otras mejoras.

El uso de puntos de referencia no es un método que se haya explorado mucho en el campo de reconstrucción 3D, dentro este trabajo se aplican algoritmos necesarios para la identificación de los puntos de referencia sobre las imágenes, permitiendo con estos realizar la alineación de nubes de puntos y finalmente hacer la reconstrucción de la malla 3D. La metodología empleada resumida se puede observar en la Figura 1.

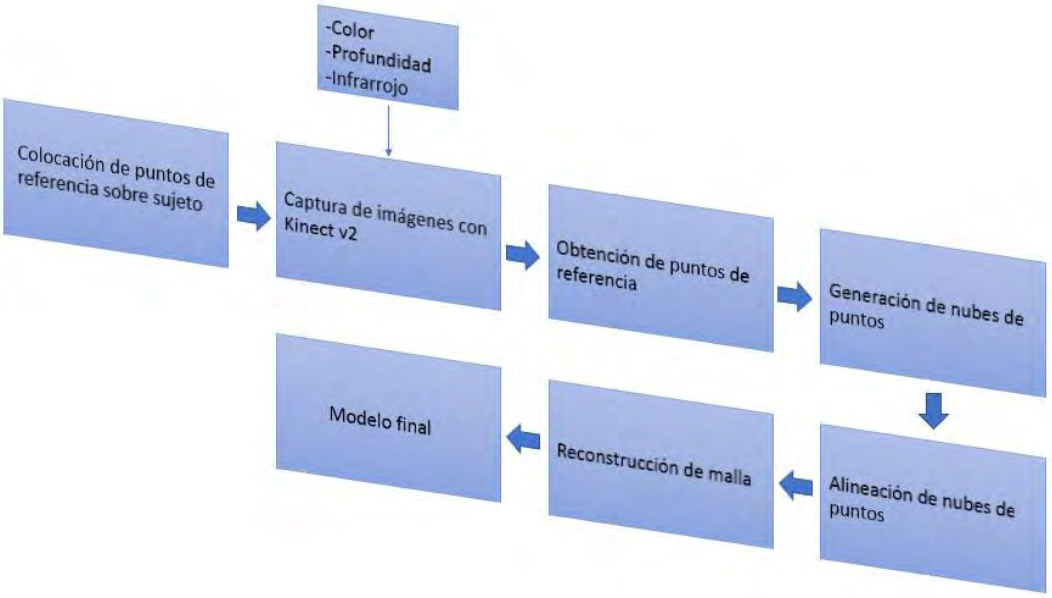

Figura 1 Metodología resumida en los pasos para obtención de modelo 3D.

[1] https://www.artec3d.com/portable-3d-scanners/artec-spider [Acceso: 10-Mar-2018]
[2] https://3dprint.com/179170/3d-scanner-buying-guide-2017 [Acceso: 10-Mar-2018]
[3] https://www.aniwaa.com/best-3d-scanner [Acceso: 10-Mar-2018]
[4] https://developer.microsoft.com/en-us/windows/kinect [Acceso: 10-Mar-2018]

El sistema se conforma de uno o dos Kinect v2, los puntos de referencia que se colocan en el paciente, un equipo de cómputo capaz de soportar los Kinect y las aplicaciones empleadas: la aplicación para la adquisición de imágenes, el software desarrollado dentro de MATLAB para la obtención y alineación de nube de puntos y un software de modelado para la reconstrucción de mallas 3D.

El presente trabajo explora la posibilidad de hacer un método accesible para la reconstrucción 3D del cuerpo humano. El trabajo se divide de la siguiente forma: primero se abarcan los antecedentes que existen sobre la reconstrucción 3D y los trabajos que han trabajado en la reconstrucción 3D con puntos de referencia, a continuación, se hace una breve descripción de los instrumentos que se ocuparon, para después hablar de los métodos de calibración que se realizaron, en seguida se habla del registro rígido y del algoritmo ICP, posteriormente se habla de los experimentos realizados y los resultados obtenidos. Por último, la conclusión obtenida y el trabajo a futuro.

1.1 Antecedentes

Dentro del área de reconstrucción 3D existen diferentes métodos que se emplean para modelar desde una escena, el cuerpo de una persona o simplemente un objeto; estos modelos se llegan a utilizar en distintos tipo de aplicaciones, por ejemplo, simuladores en medicina, sistemas de navegación, telepresencia, entre otras. En este capítulo se abordara solamente los trabajos relacionados a la reconstrucción del cuerpo humano con el Kinect v1 y v2.

Las cámaras RGB-D nos permiten capturar imágenes RGB y la información de profundidad, con la introducción del Kinect, a comparación de los equipos profesionales, mantiene un bajo costo, fácil uso y permite obtener imágenes de calidad. Por lo tanto, también se han realizado trabajos por investigadores usando el Kinect v1 y v2 para la reconstrucción 3D.

En [1] desarrollaron un método para la reconstrucción del cuerpo humano mediante un arreglo de tres Kinects v2, dos de ellos se calibran con respecto al tercero, es decir, estiman la posición de los Kinect para luego obtener las matrices de transformación de dos cámaras a la cámara de referencia. Las transformadas se aplican respectivamente a los puntos conseguidos de cada una de las cámaras.

También en [2] emplean un arreglo de Kinects v2 para escanear diferentes posiciones (central, superior e inferior) por cada vista, la persona se coloca una plataforma giratoria para conservar la misma distancia.

Dentro de los trabajos de [3]–[5] utilizan un solo Kinect v1, para [3] por separado escanean la cabeza y el cuerpo. La cabeza se escanea en diferentes poses tomando como referencia una pose frontal, para el cuerpo la persona debe girar en su posición para tomar 4 posiciones principales, por último se realiza el registro para unir ambos modelos de la cabeza y cuerpo.

En [4] la persona va a estar rotando para capturar 6 vistas. En cada vista se toman 3 capturas (horizontal, superior e inferior) haciendo un total de 18 capturas para la reconstrucción. Por cada una vista se alinea la información y por último se hace el registro de todas las vistas.

En [5] el Kinect v1 se encuentra alrededor de 2[m] y la persona debe girar 360 grados, el método que realizaron aplica una transformada rígida y no rígida para obtener mejores resultados en la alineación, la nube de puntos final se aplica la reconstrucción de malla de Poisson.

Para [6], [7] solo reconstruyen la parte frontal del cuerpo, para [6] colocan el Kinect v1 a una distancia de 1.5 [m] para una adquisición óptima, en esta posición no capturan todo el cuerpo humano por lo que se capturan 3 vistas en diferentes ángulos. En su metodología reducen las nubes de puntos antes de emplear el algoritmo ICP (*Iterative Closest Point*) para después hacer el registro entre ellas.

En el enfoque de [7] solo capturan la parte superior del cuerpo, las nubes de puntos las ponen sobre un mismo sistema de referencia para luego aplicar el algoritmo ICP.

En [8] utilizan dos Kinects v1 en posición opuesta (uno frente al otro), emplean una pieza de papel como objeto geométrico de calibración. Igual que en [3], [4], [6] Por medio del motor del Kinect v1 toman 3 tomas por cada uno a diferentes ángulos. Proceden hacer el alineamiento de nubes de puntos y rellenan la brecha perdida por medio de curvas de Bézier.

El trabajo de [9] recurre a seis Kinects v1 dirigidos hacia la persona alrededor de una circunferencia, las imágenes RGB-D se capturan simultáneamente. A partir de esto su metodología remueve la información no deseada para después hacer el registro entre las vistas vecinas por medio de la calibración estéreo y por ultimo realiza un registro global de las vistas parciales para la nube de puntos final.

1.2 Trabajos relacionados a la reconstrucción 3D por medio de puntos de referencia

Al acercamiento de este trabajo requiere de una persona que coloque los puntos de referencia en diferentes posiciones del cuerpo del sujeto a reconstruir. Actualmente, a diferencia del presente trabajo, dentro de los trabajos revisados los puntos de referencia que se llegan a utilizar son puntos de referencia digitales detectados y no son utilizados para hacer reconstrucción 3D del cuerpo humano, por lo que no se hablará sobre ellos en este trabajo.

1.3 Objetivo

La reconstrucción de un modelo 3D detallado del cuerpo humano a partir de puntos de referencia y del sensor Kinect v2.

2 Instrumentos

En este capítulo se explican los instrumentos que se emplearon, el software y el hardware necesario, así como los puntos de referencia que se utilizaron de este trabajo.

2.1 Hardware

Para este trabajo se emplea la cámara Kinect v2 que es la segunda versión desarrollada por Microsoft. Cuenta con las siguientes características[5]:

Tabla 1 Caracteristicas del Kinect v2.

Característica	Descripción
Seguimiento corporal	Puede seguir hasta 6 esqueletos completos (comparado a dos del sensor original) y 25 articulaciones (comparado a 20 con el sensor original
Detección de profundidad 512 x 424 30 Hz FOV: 70 x 60	
1080p camera de color 30 Hz	
Capacidades Infrarrojas (IR) 512 x 424 30 Hz	
Dimensiones (Longitud x ancho x altura)	24.9 cm x 6.6 cm x 6.7 cm Aproximadamente 2.9m de largo en el cable Aproximadamente 1.4 kg de peso Sensor FOV 70 x 60
Multi-arreglo de micrófonos	Cuatro micrófonos para capturar sonido, guardar audio, y encontrar la fuente de sonido y la dirección de la onda de sonido.

[5] https://developer.microsoft.com/en-us/windows/kinect/hardware

Se empleó un equipo de cómputo capaz de soportar el Kinect v2 con la configuración de hardware recomendada por Microsoft con las siguientes especificaciones[6]:

Tabla 2 Especificaciones mínimas para el soporte del Kinect v2.

Windows 8
Procesador 64 bit (x64)
Memoria RAM 4 GB
I7 3.1 GHz
Controlador de host USB 3.0 incorporado
Adaptador gráfico capaz de usar DX10

Desde este momento se hace referencia al Kinect v2 solamente como Kinect, en caso que se deba hacer una diferencia entre las versión del Kinect v1 y Kinect v2 se especificara la versión.

2.2 Software

A continuación se describe brevemente el software que se empleó:

Matlab 2017b[7] - Herramienta generalmente utilizada para el desarrollo de software matemático. Bajo esta herramienta se desarrolló el software para el procesamiento de imágenes, la obtención de los puntos de referencia sobre las imágenes, la generación de nubes de puntos y el registro de las nubes de puntos. Nos permite obtener los parámetros intrínsecos y extrínsecos de la cámara por medio de la calibración.

Meshlab[8] - Software de uso libre que permite el manejo de mallas. Este software se empleó para la reconstrucción de la malla del modelo 3D de la nube de puntos final por medio de la reconstrucción de mallas de Poisson.

Multiple Kinect Capture - Software desarrollado en el Instituto de Ciencias Aplicadas y Tecnología (ICAT, UNAM) para la captura de las imágenes RGB, infrarrojo (IR) y profundidad por medio de

[6] https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn782036(v%3dieb.10)
[7] https://www.mathworks.com/products/matlab.html
[8] http://www.meshlab.net/

uno o varios Kinects. Para el uso de los Kinects simultáneos se empleó la librería libfeenect[9]. En este trabajo solo se emplearon hasta dos Kinects.

A fin de emplear dos Kinects de manera simultánea se requiere que el equipo a contenga dos puertos USB 3.0.

Fiji[10] – Programa de uso libre para el procesamiento de imágenes digitales. Se utilizó para el desarrollo del experimento del capítulo 3.3.2.

2.3 Puntos de referencia

Un punto de referencia se define en este trabajo como: Un objeto que se emplea y se detecta relativamente fácil en la escena, para establecer un punto de referencia espacial entre las imágenes capturadas por un sistema (en este caso el Kinect).[11]

A partir de este capítulo los puntos de referencia se le harán referencia como *landmark* (del inglés) en el resto del trabajo.

Los landmarks que se utilizan están elaborados por un material adherible a la piel y cinta reflejante (Figura 2), tienen un diámetro alrededor de 0.5 [cm]. Por medio de la cinta reflejante, los puntos de referencia producen un punto ciego en el sensor (emisor) de infrarrojo, estos puntos ciegos captados se observan en la imagen de infrarrojo obtenida de la captura de imágenes realizada con el Kinect.

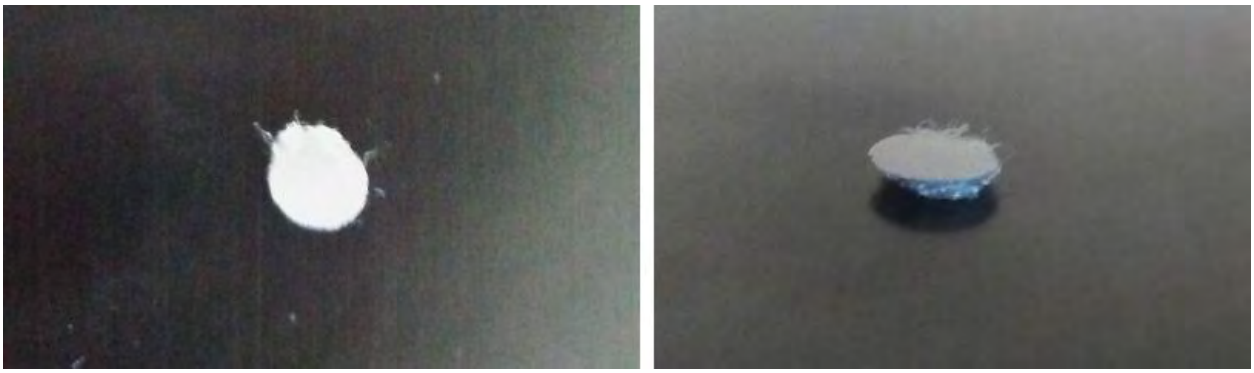

Figura 2 Landmark elaborado por material adherible a la piel y cinta reflejante.

Estos puntos de interés se toman como referencia para alinear las nubes de puntos que son generadas durante el procesamiento de imágenes, con esto se logra que se necesite solo el número de puntos de referencia utilizado en el alineamiento y no requiera el uso completo de puntos que contiene la nube de puntos.

[9] https://github.com/OpenKinect/libfreenect
[10] https://fiji.sc/
[11] La definición se hace en base de la descripción del marcador fiducial de [10].

3 Calibración

El lente de una cámara puede llegar a generar distorsión en las imágenes que se toman. La distorsión llega a deformar las imágenes tal que puede afectar al modelo que se desea generar, la calibración es el proceso por el cual se hace la corrección de este error que se produce. En la reconstrucción 3D este procedimiento es de importancia porque de las imágenes 2D se extrae la información necesaria y de esta manera se remueve el error.

Además, como se observa en la Figura 3 la información capturada entre las imágenes no llega a ser precisamente la misma debido a que capturan distintos planos e igualmente entre los tipos de imágenes la resolución difiere en el Kinect, la de RGB tiene una resolución de 1920x1080 y la de profundidad e infrarrojo es de 512x424.

Figura 3 Comparación entre imagen RGB, IR y profundiad.

Por lo tanto, en las cámaras RGB-D (como el Kinect), se necesita encontrar los puntos coincidentes entre las capturas de color y profundidad, es decir, se deben alinear las imágenes en un mismo plano. Este procedimiento se le conoce como registro de imagen y para este paso se requiere conocer los parámetros intrínsecos y extrínsecos de la cámara que se obtienen por medio de la calibración.

Los parámetros intrínsecos son las características internas de la cámara, permiten pasar de las coordenadas 3D del mundo a las de coordenadas 2D de la imagen. Los parámetros son el centro óptico que en el mejor de los casos es el centro de la imagen y la longitud focal que es la distancia que existe entre el plano focal y el plano de la imagen.

Los parámetros extrínsecos definen la posición y orientación de la cámara en coordenadas del mundo, se compone de una traslación y una rotación.

3.1 Calibración Estéreo

En los sistemas de estereovisión (sistemas comprendidos de múltiples cámaras) como en [1], [2], [9], un mismo punto visto por N cámaras llega a tener una diferente posición en las imágenes por la diferente perspectiva en la que fue proyectado (Figura 4). Esta diferencia que existe se le conoce como disparidad, de modo que en estos sistemas se debe estimar la posición de cada una de las cámaras que se emplea, permitiendo transformar la posición de un punto en un sistema de referencia a otro sistema de referencia distinto.

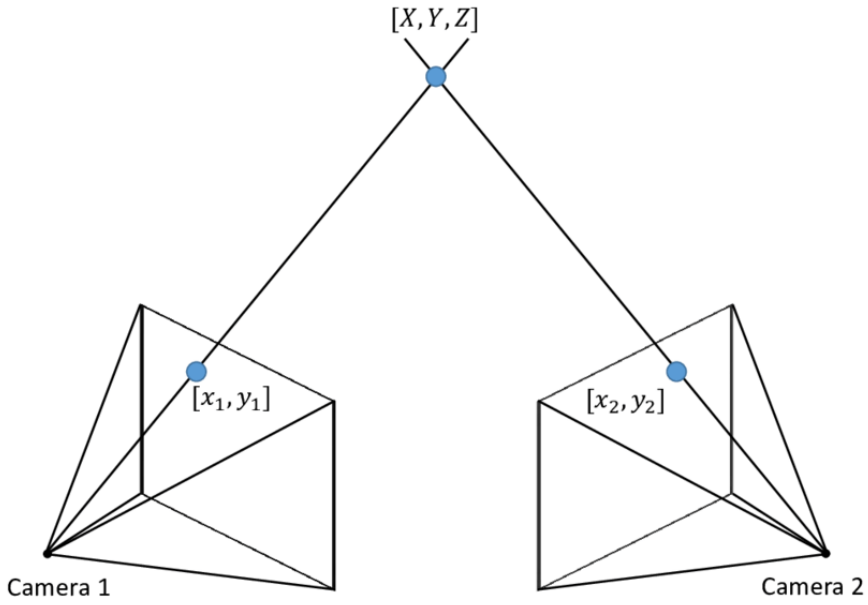

Figura 4 Un mismo punto observado por dos cámaras en distinta posición. El punto tiene diferentes coordenadas en cada una de las cámaras por la diferente posición y orientación de la cámara. Imagen tomada de [11].

Por medio de la calibración estéreo se puede encontrar la posición de cada dispositivo permitiendo transformar el sistema de referencia de cada cámara a un solo sistema de referencia.

3.2 Distorsión

Los tipos de distorsión más frecuentes son la distorsión tangencial y la distorsión radial, estas generan cierta deformación en la imagen ocasionando que se obtengan datos erróneos en la reconstrucción.

La distorsión tangencial ocurre cuando el plano del lente y el de la imagen no son paralelos, sucede por la mala alineación que tiene el lente y el sensor.

La distorsión radial llega ser más común y ocurre cuando se curva la imagen, usualmente se observa en los bordes de la imagen.

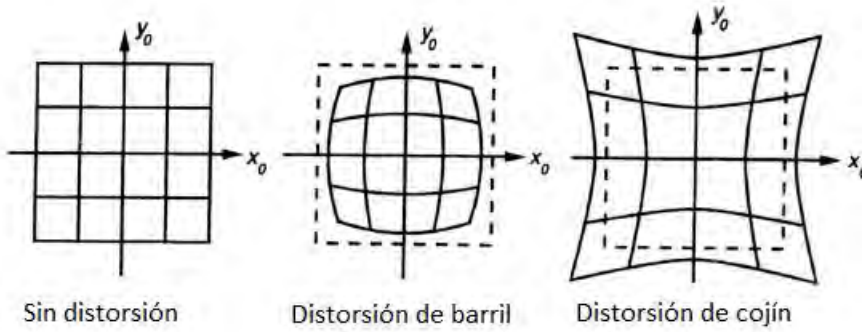

Figura 5 Ejemplo de distorsión radial. Imagen tomada de [12].

3.3 Método de calibración

Para la corrección de la distorsión que se genera en las imágenes RGB e IR se emplea el método clásico de [13] y como previamente se mencionó, la resolución de las imágenes y la posición de las cámaras entre las de RGB e IR es diferente por lo que cada uno se calibra por separado. Por otra parte, para las imágenes de profundidad no se puede utilizar este método debido a que no es posible detectar el patrón de calibración, de modo que se genera una función mediante un experimento parecido al desarrollado en [14] para calibrar la profundidad.

3.3.1 Calibración de imágenes IR y RGB.

En este método se pone sobre una superficie plana un patrón gráfico de calibración y con el Kinect se toman imágenes RGB e IR desde diferentes puntos de vista en los que se debe ver el patrón completo (como se ve en la Figura 6 y Figura 7), el procedimiento corrige la deformación curva enderezando las líneas del patrón a su forma original. El patrón utilizado fue de 8x6 cuadros con 50 mm cada uno.

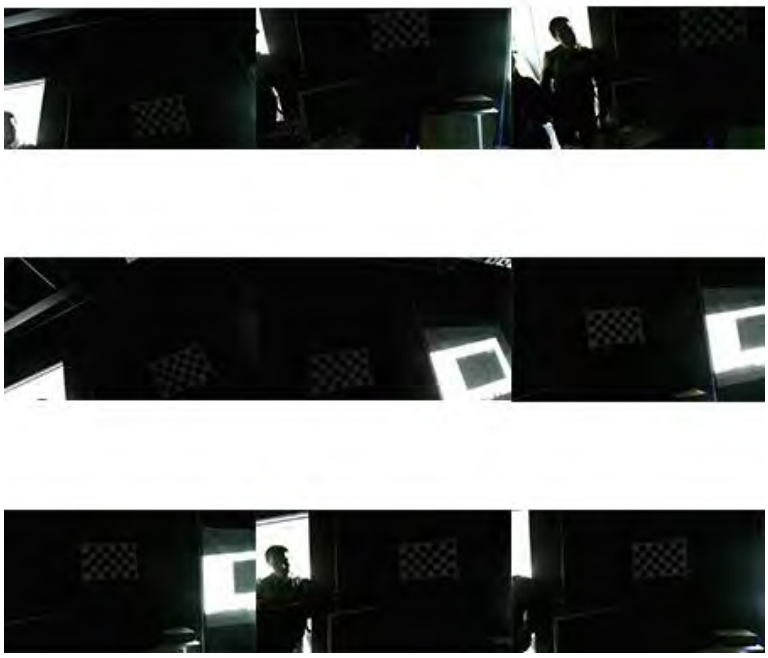

Figura 6 Nueve diferentes puntos de vista con imágenes RGB.

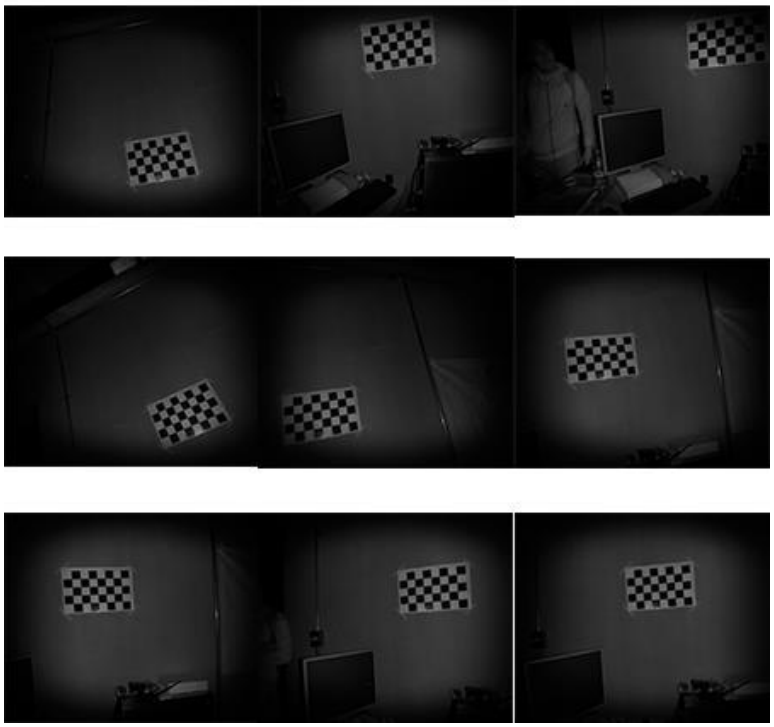

Figura 7 Nueve diferentes puntos de vista con imágenes IR.

Para obtener los parámetros intrínsecos de la cámara de RGB e IR se seleccionan por medio de la aplicación de MATLAB "Camera Calibrator" las imágenes correspondientes que se tomaron por separado, en este trabajo se emplean 15 vistas diferentes para cada una de las cámaras. La herramienta debe detectar en todas las imágenes el patrón de calibración, este mismo programa descarta las imágenes en las que no lo detecta, con las imágenes útiles se realiza la calibración y se obtienen los siguientes resultados para el caso de un Kinect:

Tabla 3 Parámetros intrínsecos (IR)

Longitud focal	$(374.511 \quad 375.002)$
Centro Óptico	$(256.063 \quad 210.879)$

Tabla 4 Parámetros intrínsecos (RGB)

Longitud focal	$(1074.926 \quad 1076.313)$
Centro Óptico	$(936.683 \quad 545.747)$

Se almacena la información de los parámetros intrínsecos que MATLAB genera con los resultados obtenidos por cada uno de los tipos de imágenes (RGB e IR).

Al obtener los parámetros intrínsecos después se buscan los parámetros extrínsecos por medio de MATLAB, los siguientes resultados fueron obtenidos para el caso de un Kinect:

Tabla 5 Parámetros extrínsecos (IR)

Vector de translación	$(256.811248619987 \quad -127.647590216958 \quad 1184.62215044559)$
Matriz de rotación	$\begin{bmatrix} -0.994690689162131 & 0.0760499042346218 & 0.0693314139482943 \\ -0.0791874894280434 & -0.995902046508980 & -0.0436858704538499 \\ 0.0657249907740461 & -0.0489441091974493 & 0.996636693967576 \end{bmatrix}$

Tabla 6 Parámetros extrínsecos (RGB)

Vector de translación	$(291.128886987149 \quad -118.572144112579 \quad 1242.79162395873)$
Matriz de rotación	$\begin{bmatrix} -0.994035775617613 & 0.0759515069310990 & 0.0782575580195731 \\ -0.0784873964891265 & -0.996468101386525 & -0.0298504859506763 \\ 0.0757139708683305 & -0.0358146829390962 & 0.996486178078413 \end{bmatrix}$

Con la finalidad de obtener las imágenes sin deformaciones, por cada tipo de imagen (RGB e IR) se utiliza respectivamente el objeto generado previamente con MATLAB para obtener las imágenes corregidas sin distorsión.

Por lo tanto, se puede apreciar en la Figura 8 y Figura 9 los resultados obtenidos aplicados en las imágenes.

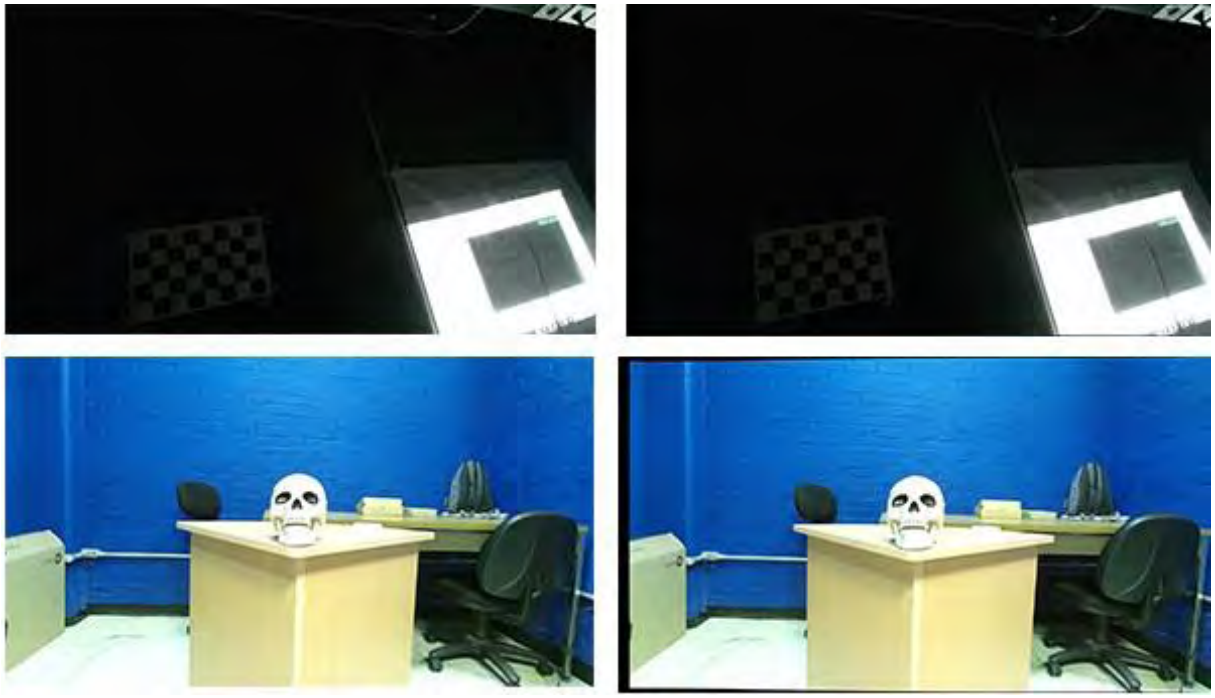

Figura 8 Imágenes RGB. Las imágenes de la izquierda son las que se obtienen inicialmente y tienen distorsión, las imágenes de la derecha se remueve la distorsión.

Figura 9 Imágenes IR. Las imágenes de la izquierda son las que se obtienen inicialmente y tienen distorsión, las imágenes de la derecha se remueve la distorsión.

3.3.2 Calibración de profundidad

En el caso de las imágenes de profundidad, para eliminar la distorsión, se utiliza la distancia sin procesar por cada pixel en la imagen y se transforma a metros mediante una función que obtenemos al desarrollar el siguiente experimento:

Se necesita el Kinect, una mesa, un medidor de distancia laser y un objeto con superficie plana, en este caso se utilizó una caja. El Kinect lo colocamos a una distancia de 60-75 [cm] de la mesa, este debe estar lo más recto posible y para no modificar su posición se usa un tripié. El objeto se coloca sobre la mesa en frente del Kinect a una distancia donde se pueda captar sin ruido, el láser se coloca sobre el sensor examinando que sea capaz de registrar la distancia del objeto en su posición inicial y a lo largo de la mesa, teniendo en cuenta que no se debe modificar su posición. En la Figura 10 se ve el arreglo del equipo.

Figura 10 Arreglo del equipo para calibracion de profundidad.

El objeto se va recorriendo hacia atrás 2.5 [cm] hasta llegar al límite de la mesa, por cada nueva posición, se registra la distancia que indica el láser y con el Kinect se obtiene la imagen de profundidad donde se obtiene el valor promedio del histograma de una área seleccionada.

Para obtener el valor promedio de las imágenes de profundidad se utiliza el software Fiji. Con este se selecciona la misma área en todas las imágenes (Figura 11), el área tiene que cubrir solamente el objeto por lo tanto se toma la última imagen capturada (la que está a mayor distancia) y se cubre la mayor parte posible del objeto para obtener el valor promedio por medio del histograma como se ve en la Figura 12.

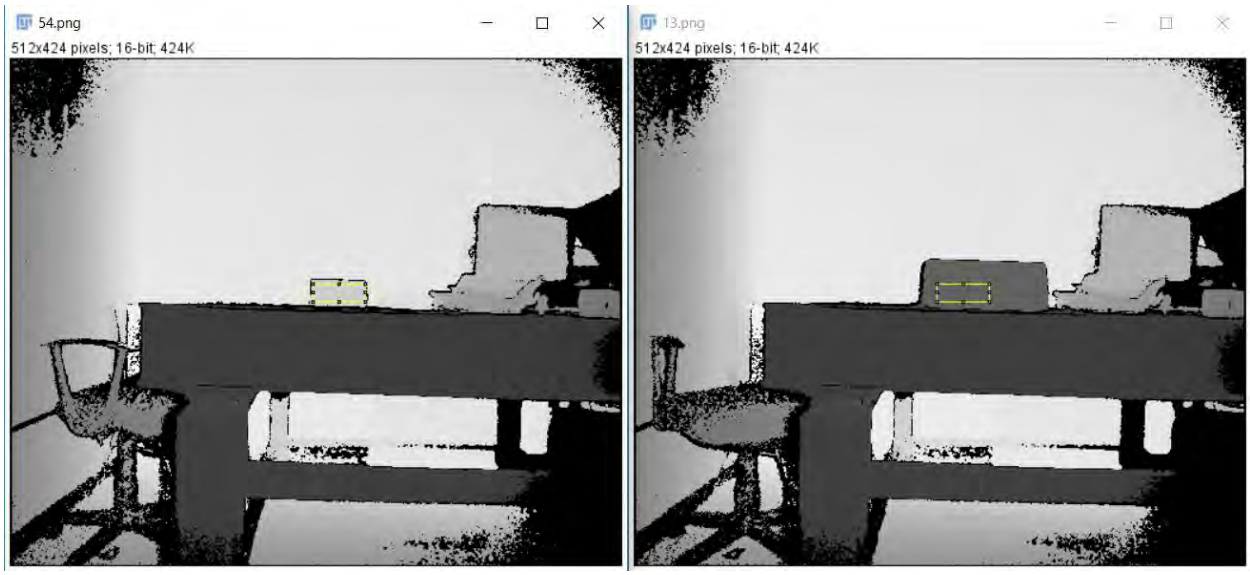

Figura 11 La imagen de la derecha es la primera distancia y la de la izquierda es la última distancia en la que se colca el objeto, todas las imágenes tienen la misma área seleccionada.

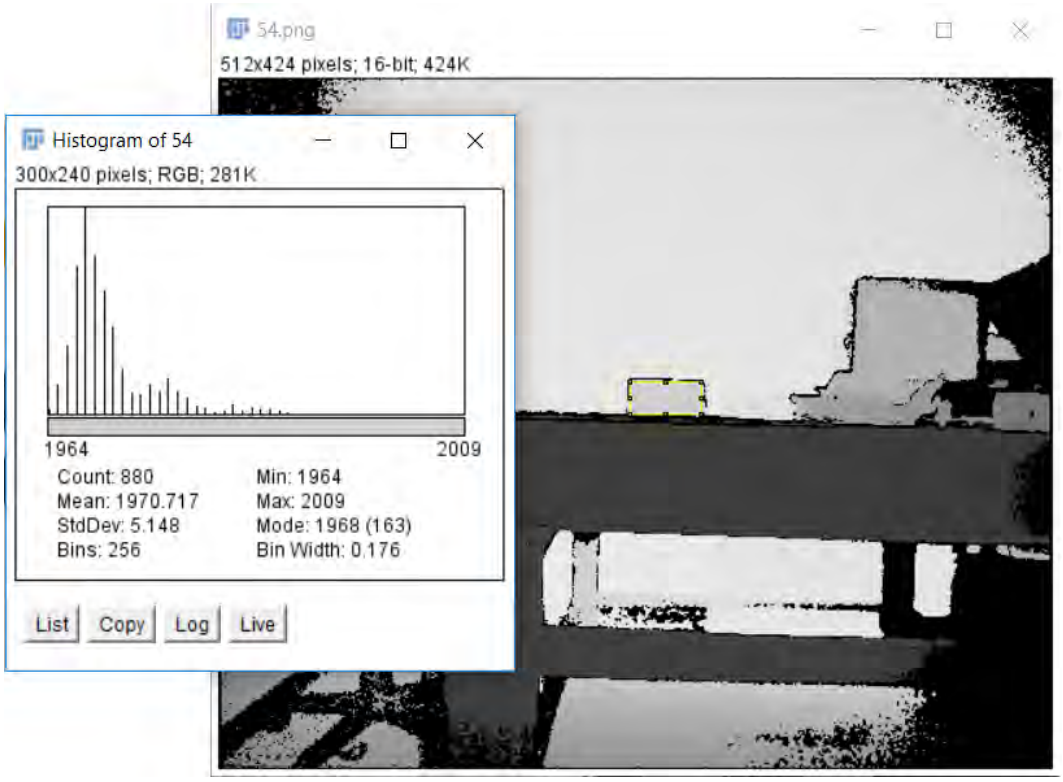

Figura 12 Obtención del valor promedio de área seleccionada.

Para cada una de las imágenes que se tomó se obtiene el valor promedio del histograma (que corresponde a la distancia que registro el láser) de la misma área seleccionada en la última imagen.

Para el experimento se obtuvieron 42 datos con los que se obtuvo la Tabla 7 y en la Figura 13 se ve la gráfica obtenida de estos datos.

Tabla 7 Datos obtenidos en el experimento

Valor promedio del área	Valor promedio del área en metros	Distancia registrada por láser en metros
929.05	0.99795	1.0017125
955.039	1.023939	1.025525
980.67	1.04957	1.050925
1006.888	1.075788	1.076325
1033.164	1.102064	1.101725
1061.169	1.130069	1.127125
1086.328	1.155228	1.152525
1110.054	1.178954	1.1763375
1133.202	1.202102	1.2017375
1157.382	1.226282	1.2271375
1180.737	1.249637	1.254125
1204.4	1.2733	1.2811125
1228.628	1.297528	1.3065125
1255.046	1.323946	1.3287375
1280.982	1.349882	1.355725
1306.249	1.375149	1.381125
1330.826	1.399726	1.4049375
1355.683	1.424583	1.4303375
1383.826	1.452726	1.4589125
1406.87	1.47577	1.482725
1432.47	1.50137	1.508125
1458.104	1.527004	1.5351125
1483.771	1.552671	1.5605125
1508.891	1.577791	1.5859125
1532.867	1.601767	1.6113125
1558.873	1.627773	1.6367125
1582.948	1.651848	1.6621125
1610.506	1.679406	1.6875125
1638.174	1.707074	1.7129125
1664.302	1.733202	1.7383125
1690.499	1.759399	1.7637125
1715.187	1.784087	1.787525

1743.011	1.811911	1.8145125
1768.87	1.83777	1.8399125
1792.699	1.861599	1.8653125
1816.611	1.885511	1.889125
1841.815	1.910715	1.9161125
1868.146	1.937046	1.9415125
1893.924	1.962824	1.9669125
1921.65	1.99055	1.9923125
1945.171	2.014071	2.0177125
1969.327	2.038227	2.03835

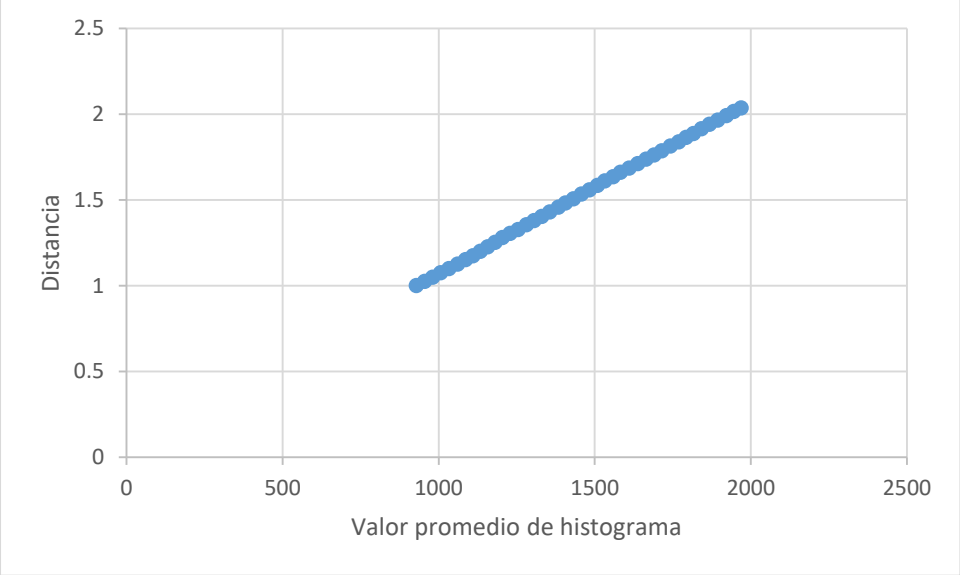

Figura 13 Grafica de los datos obtenidos durante el experimento.

Por lo tanto, la función para calibrar los datos de profundidad al no ser lineal, se obtiene aproximando una curva de calibración por medio de mínimos cuadrados:

$$Distancia\ en\ metros = 0.001 * (Intensidad\ de\ gris\ del\ pixel) + 0.0689$$

en donde la distancia en metros de cada pixel va a ser dependiente de la intensidad de gris que tenga.

3.3.3 Mapeo de color

Una nube de puntos se genera por medio de la imagen de profundidad y se calibra la información por medio del experimento realizado en el capítulo 3.3.2, para agregarle el color correspondiente a cada punto de la nube se debe encontrar la relación que existe con la imagen RGB.

En [15] se encuentra el procedimiento que se implementó para alinear la información de las imágenes y está basado en las ecuaciones que en [16] plantean, este proceso se realiza por cada una de las vistas que se toma.

Para este algoritmo se ocupan los parámetros intrínsecos y extrínsecos de RGB e IR, la imagen de profundidad y la imagen de RGB.

La información 2D de profundidad se proyecta con respecto a la cámara IR a una posición 3D por medio de las siguientes ecuaciones:

$$X_{ir} = \frac{(x_{ir} - c_{xir}) * d(x_{ir}, y_{ir})}{f_{xir}} \tag{1}$$

$$Y_{ir} = \frac{(y_{ir} - c_{yir}) * d(x_{ir}, y_{ir})}{f_{yir}} \tag{2}$$

$$Z_{ir} = d(X_{ir}, y_{ir}) \tag{3}$$

Donde

$d(X_{ir}, y_{ir})$ es cada pixel que la imagen de profundidad tiene

c_{xir}, c_{yir} es el centro óptico de IR

f_{xir}, f_{yir} es la longitud focal de IR

Para conseguir el color de cada punto se debe proyectar las coordenadas (X_{ir}, Y_{ir}, Z_{ir}) a la imagen de RGB

$$\left(X_{rgb}, Y_{rgb}, Z_{rgb}\right) = R * (X_{ir}, Y_{ir}, Z_{ir}) + T \tag{4}$$

Existe una pequeña diferencia de posición entre la cámara RGB e IR por lo tanto la rotación R y traslación T es la relación que hay entre ellas y para esto se emplea la matriz

$$P_{rgb} = \begin{bmatrix} R_{rgb}^{-1} R_{ir} & R_{rgb}^{-1}\left(t_{ir} - t_{rgb}\right) \\ 0 & 1 \end{bmatrix} \tag{5}$$

Donde

$$\mathrm{R} = R_{rgb}^{-1} R_{ir} \tag{6}$$

$$\mathrm{T} = R_{rgb}^{-1}\left(t_{ir} - t_{rgb}\right) \tag{7}$$

Y

R_{rgb} es la matriz de rotación de la cámara RGB

R_{ir} es la matriz de rotación de la cámara IR

t_{rgb}es el vector de traslación de la cámara RGB

t_{ir} es el vector de traslación de la cámara IR

La demostración de P_{rgb} se encuentra en el apéndice de este trabajo.

Por lo tanto, para proyectar cada punto de la nube en la imagen RGB se emplean las ecuaciones:

$$x_{rgb} = \frac{X_{rgb} * f_{xrgb}}{Z_{rgb}} + c_{xrgb} \tag{8}$$

$$y_{rgb} = \frac{Y_{rgb} * f_{yrgb}}{Z_{rgb}} + c_{yrgb} \tag{9}$$

Donde

c_{xrgb}, c_{yrgb} es el centro óptico de RGB

f_{xrgb}, f_{yrgb} es la longitud focal de RGB

x_{rgb}, y_{rgb} son coordenadas 2d que se utilizan en la imagen RGB para obtener el valor de color del pixel que le corresponde al punto en la nube

En la Figura 14 se pueden ver varios ejemplos de los resultados obtenidos de nubes de puntos.

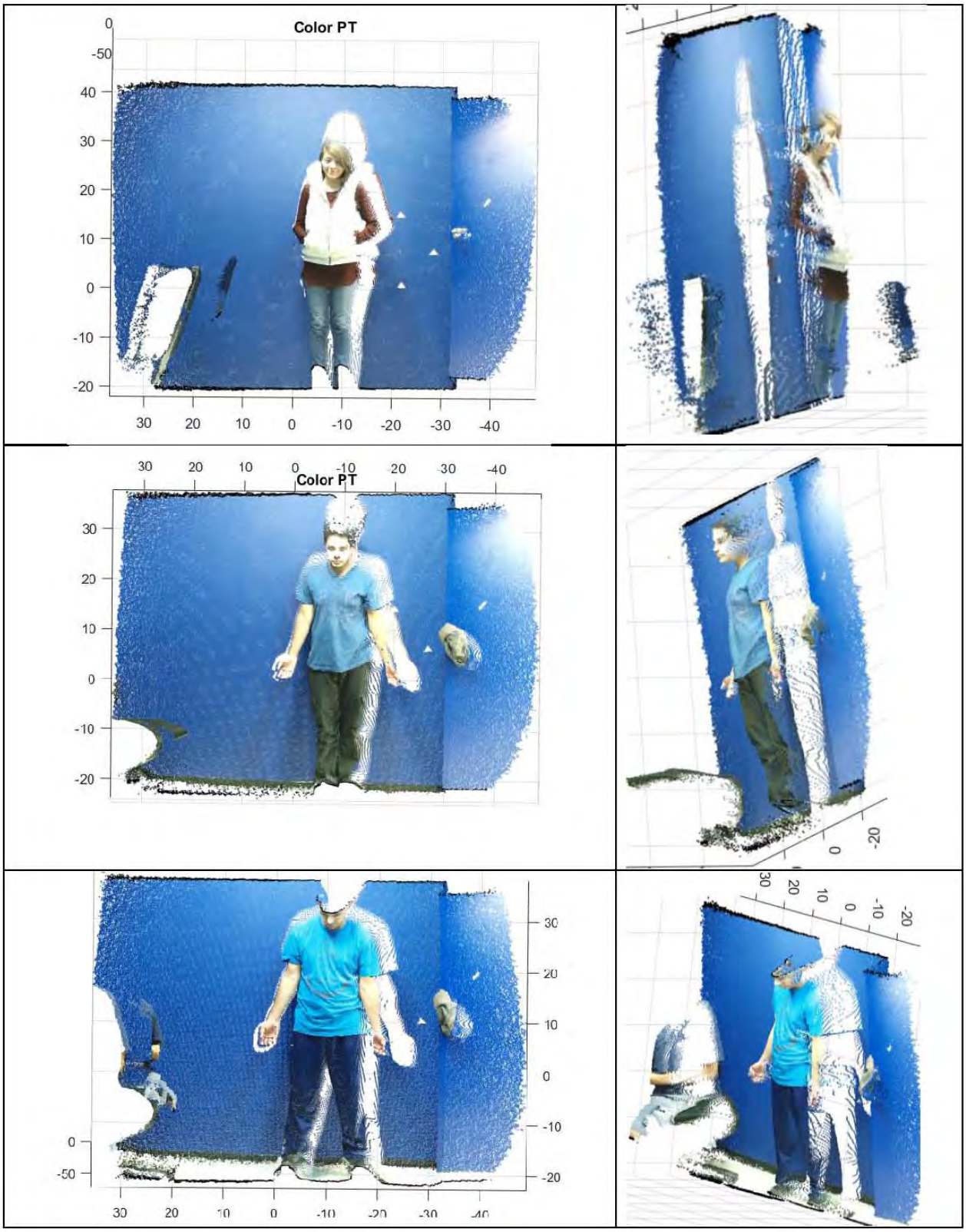

Figura 14 Ejemplos de nubes de puntos con color.

3.3.3.1 Segmentación

Al momento de generar cada una de la nube de puntos y su mapeo de color, la nube resultante tiene información que no es necesaria para el modelo final, por ejemplo, el fondo, objetos en la escena o simplemente un punto no cuenta con color.

Los métodos que se emplean en este trabajo son sencillos para poder filtrar los puntos que no se desean. Estos se explican a continuación:

Para filtrar la mayor parte de la información que no se desea se hace una detección de bordes y se aplican operaciones de morfología discreta (erosión y dilatación) sobre la imagen de profundidad, con el resultado anterior, se busca rellenar las áreas que conforman al sujeto, por medio de MATLAB se hace selección de las áreas donde se encuentre la persona como se muestra en la Figura 15. El resultado final se toma como una máscara para filtrar la información a la hora de generar la nube de puntos.

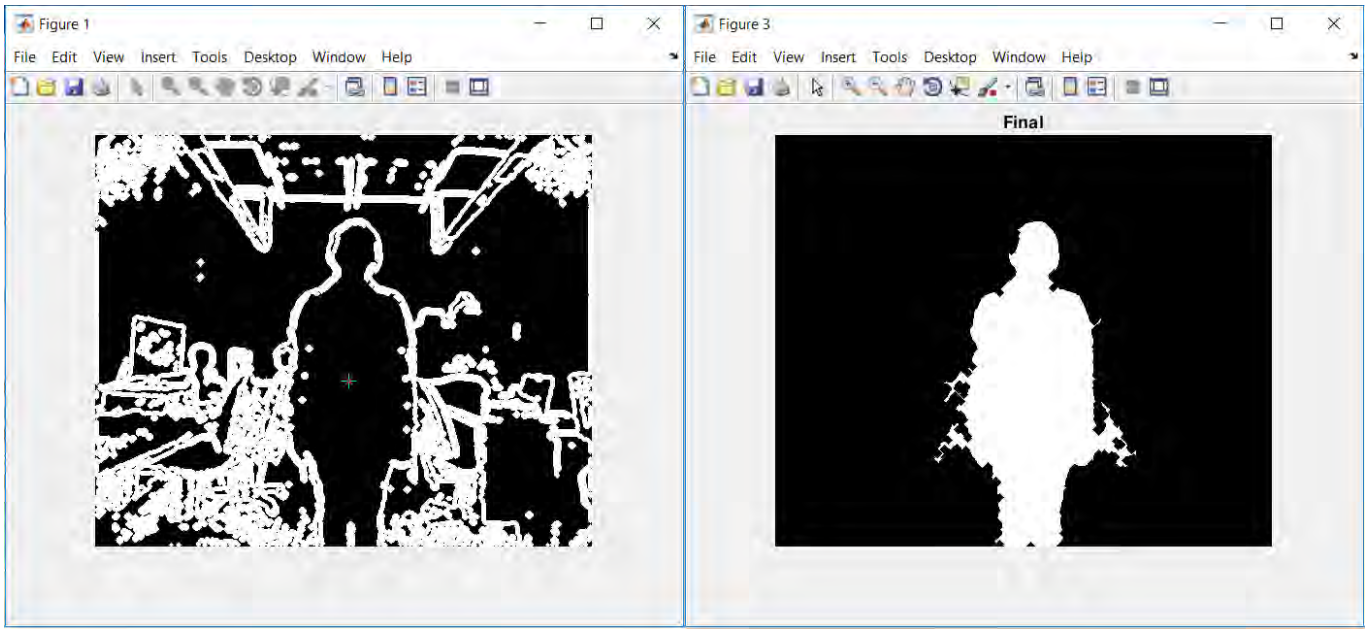

Figura 15 La imagen de la izquierda se hizo la selección el área donde se encuentra la persona (en este ejemplo, solamente el cursor que se encuentra en el centro de la persona). La imagen de la derecha es la segmentación final de la imagen de profundidad, esta se utiliza como mascara para filtrar la información al momento de generar la nube de puntos.

También se cuenta con dos umbrales, uno para la parte frontal y otra para la parte trasera de la nube de puntos para la información que no se filtró en el caso anterior. Por último, la información que no cuente con color se desecha.

Ejemplos de los resultados obtenidos se observan en la Figura 16

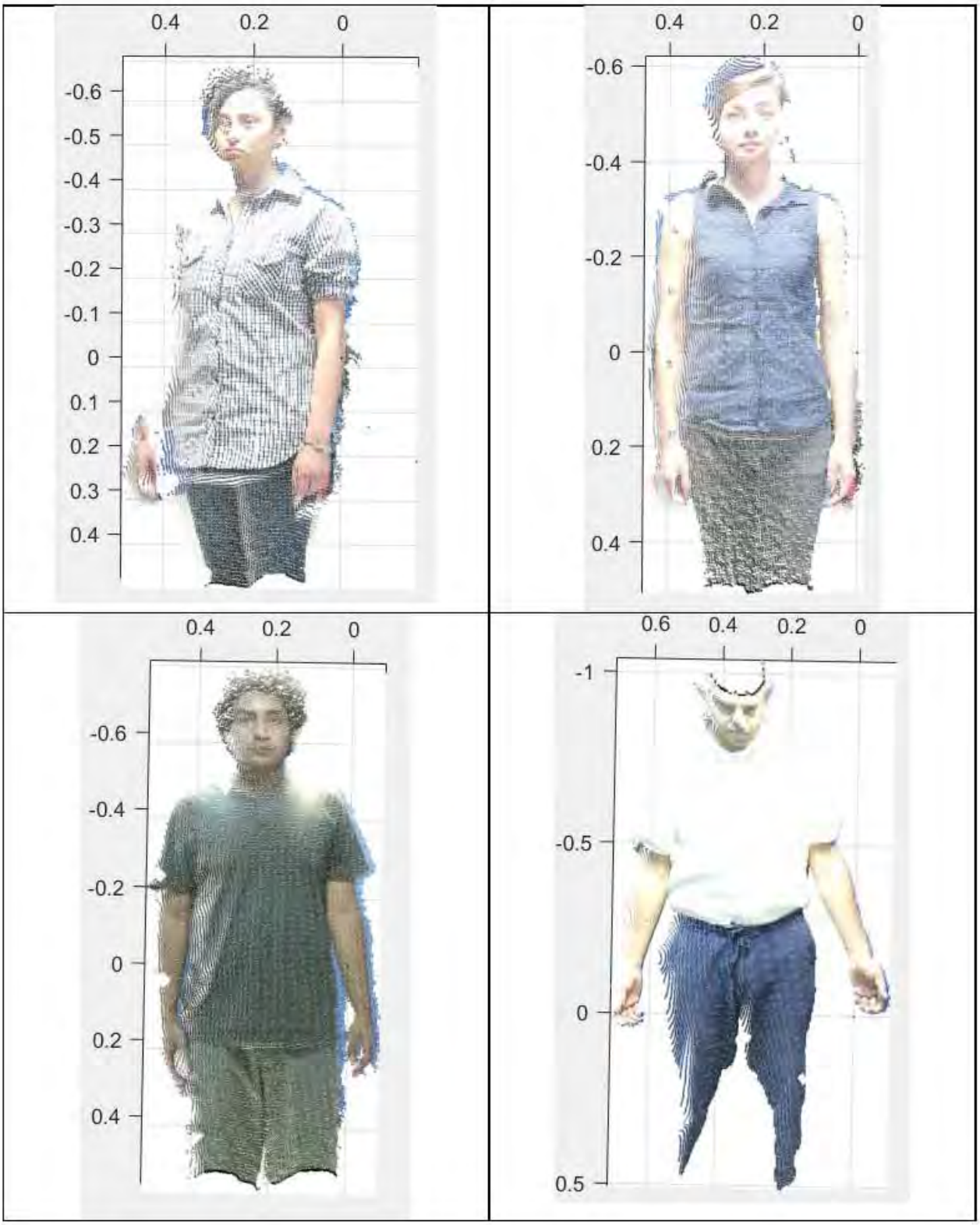

Figura 16 Ejemplos de nubes de puntos finales de cada toma.

3.4 Estimación de posición y calibración de dos cámaras frontales

Con el propósito de hacer uso de dos cámaras con posición de una frente a la otra (como se aprecia en la Figura 17) se debe encontrar el sistema de referencia de la segunda cámara con respecto a la cámara que se toma como referencia, para lograrlo se utilizan los landmarks empleados en este trabajo. Para obtener los parámetros de cada una de las cámaras se emplea el método explicado en el capítulo 3.3.1.

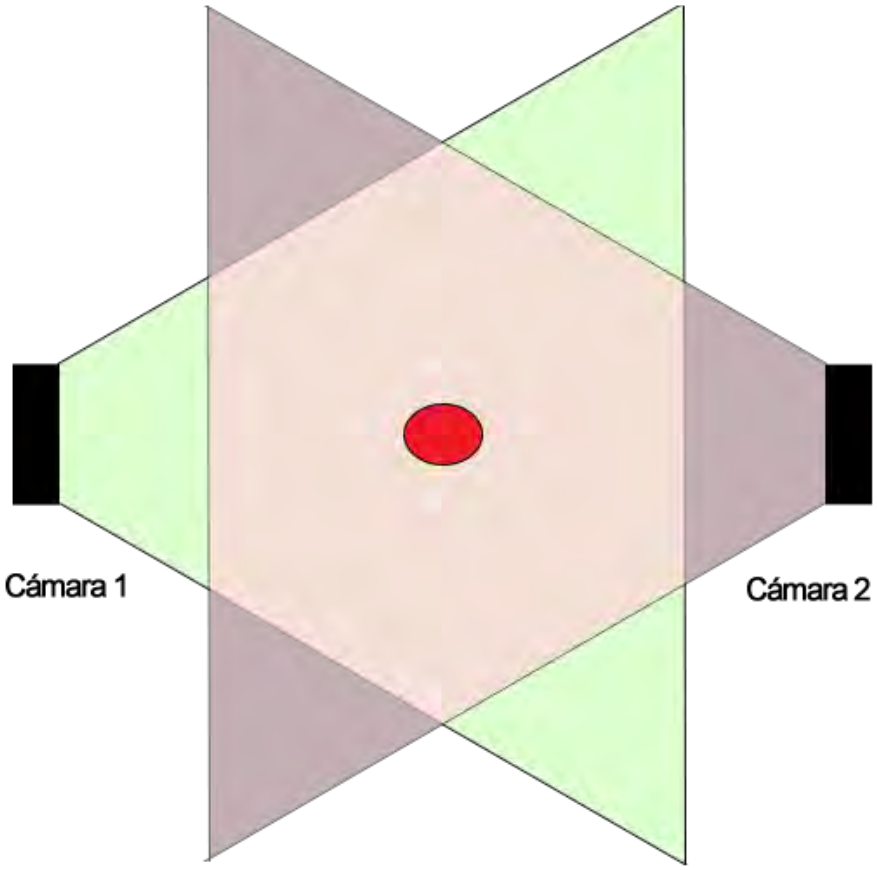

Figura 17 Posición de cámaras para este método. El punto rojo representa la posición deseada que debería tener el sujeto a la hora de capturar las imágenes.

Las cámaras se encuentran a una distancia de 3 [m] de diferencia, buscando que se encuentren frente a frente lo más aproximado posible. La cámara que se toma como referencia se coloca 4 landmarks en las esquinas del Kinect evitando colocarlos sobre las cámaras RGB e IR (Figura 18). Una vez que el sistema se encuentra listo, con la segunda cámara se obtienen las imágenes necesarias.

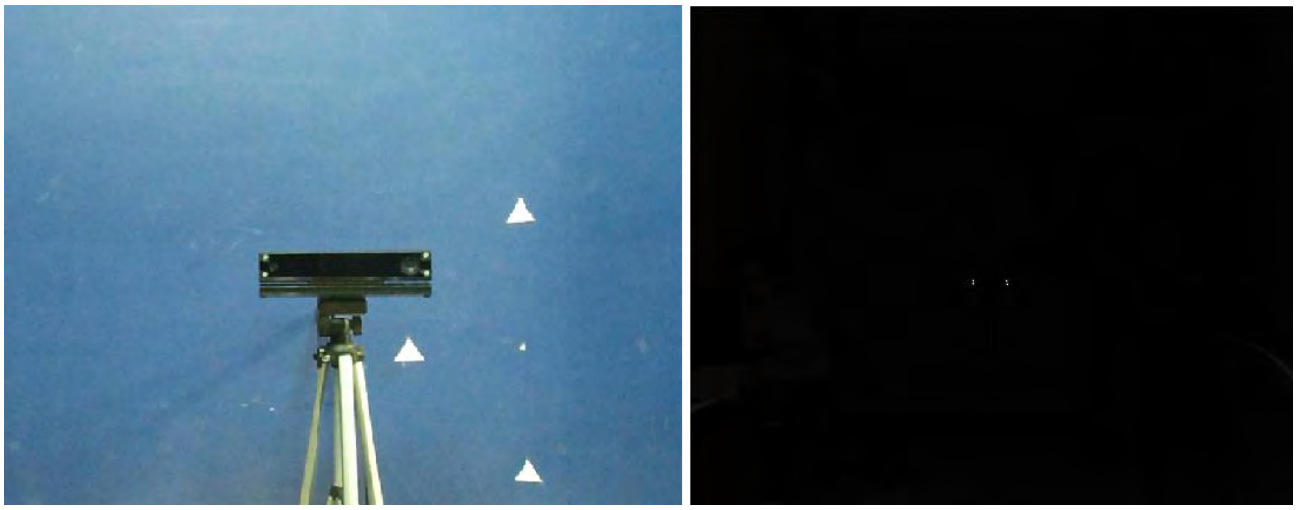

Figura 18 En la imagen RGB de la izquierda se observa la posición de los landmarks sobre las esquinas del Kinect. En la imagen IR de la derecha se observa los landmarks a detectar.

Se desea encontrar el sistema de referencia del dispositivo por medio de los landmarks en las imágenes de IR. Teniendo en cuenta el sistema original del Kinect (Figura 19) se obtienen los vectores normalizados por medio de los landmarks.

Figura 19 Sistema de referencia del Kinect v2. Imagen tomada de [17].

Los vectores obtenidos generan lo que será el sistema de referencia que representa la matriz de rotación de la cámara.

Para el vector de traslación, se obtiene el centroide de los landmarks que proporciona el valor de profundidad Z, para X y Y se estima la posición que tiene un Kinect con respecto al otro, con los siguientes resultados:

Tabla 8 Parámetros extrínsecos de la cámara que se colocaron los puntos de referencia

Vector de translación	$(0.15 \quad 0.475 \quad 3.0)$

Matriz de rotación	$\begin{bmatrix} -0.9988 & 0.0036 & -0.0036 \\ 0.0344 & 0.9994 & -0.0003 \\ 0.0036 & -0.0004 & -0.9994 \end{bmatrix}$

Una vez obtenidos la matriz de rotación y el vector de traslación, se remueven los landmarks del Kinect de referencia y se procede a obtener cada una de las dos nubes de puntos. La persona se coloca idealmente en medio de las dos cámaras (alrededor de 1.5 [m]) (Figura 20) y por medio del software de captura se toman las imágenes simultáneamente con los Kinects.

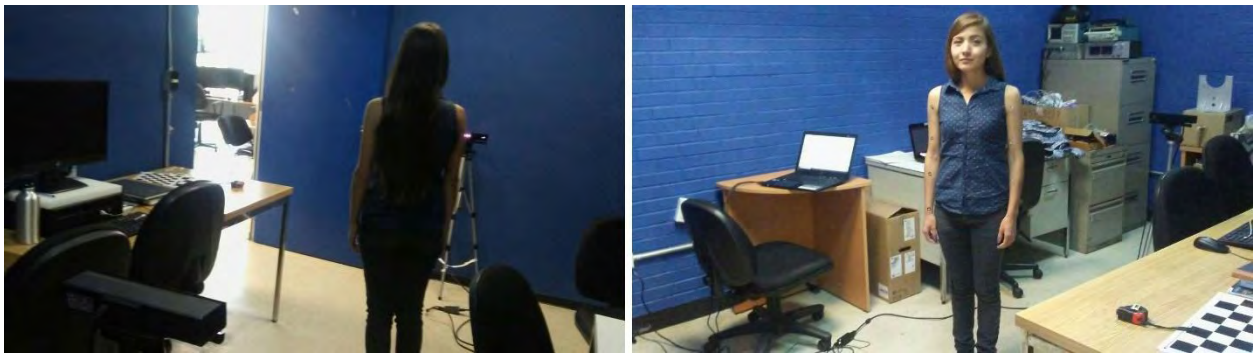

Figura 20 La persona se coloca entre las dos cámaras.

Al obtener ambas nubes de puntos, se relacionan las matrices de cámara y se aplica la transformada obtenida a la nube de puntos que fue capturada por el segundo Kinect hacia el Kinect que se tomó de referencia.

Se observa en la Figura 21 ejemplos de los resultados obtenidos.

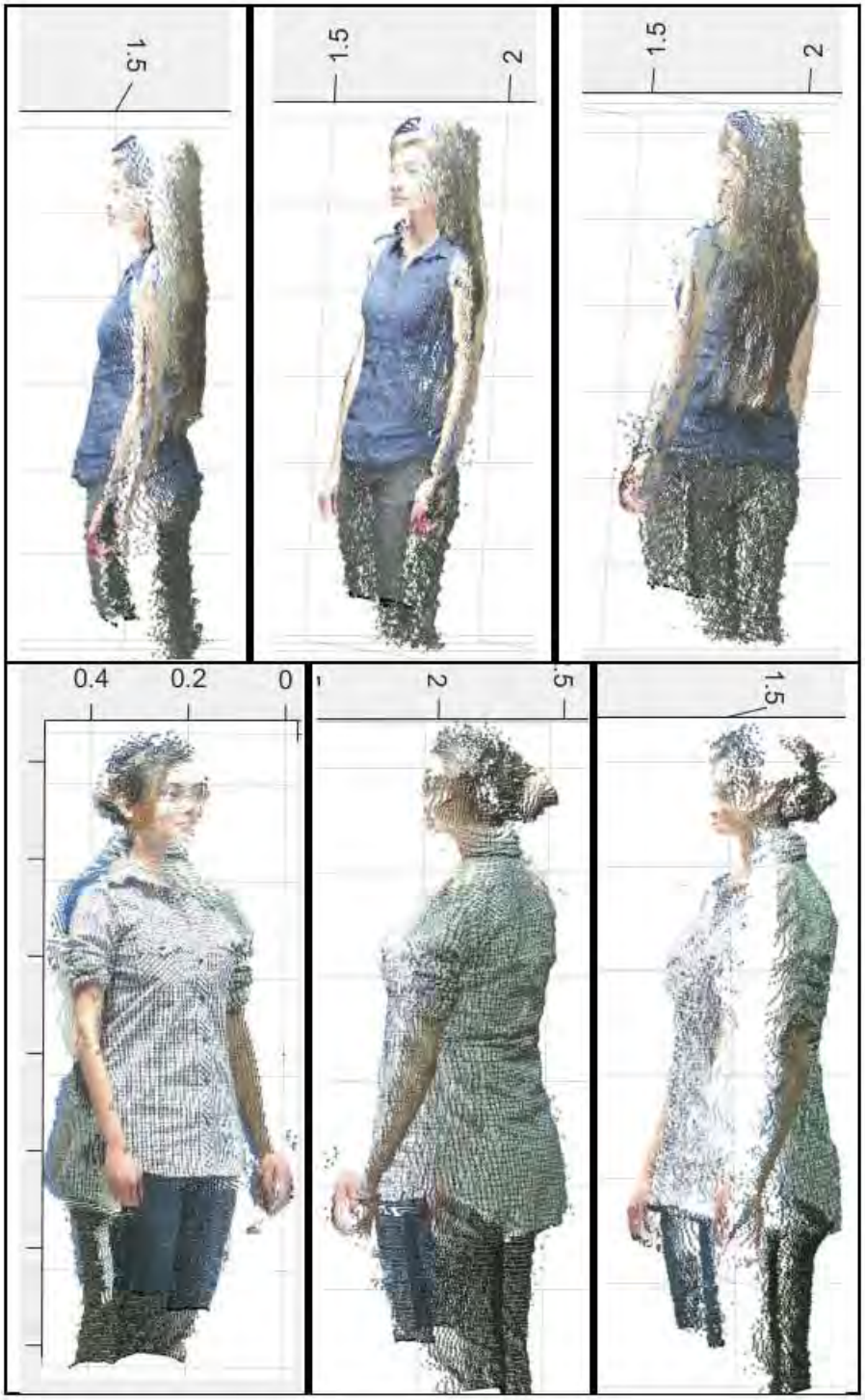

Figura 21 Ejemplos de nubes de puntos por medio de dos cámaras en posición frontal.

3.5 Estimación de posición y calibración de dos cámaras con desplazamiento lateral

En este caso las cámaras se encuentran alineadas como se ve en la Figura 22, ambas dirigidas en una misma dirección, separadas por una distancia aproximada de 1.5 [m]. El método de calibración de ambas cámaras se hace por medio de la calibración estéreo, esta se realiza con patrón de calibración previamente usado.

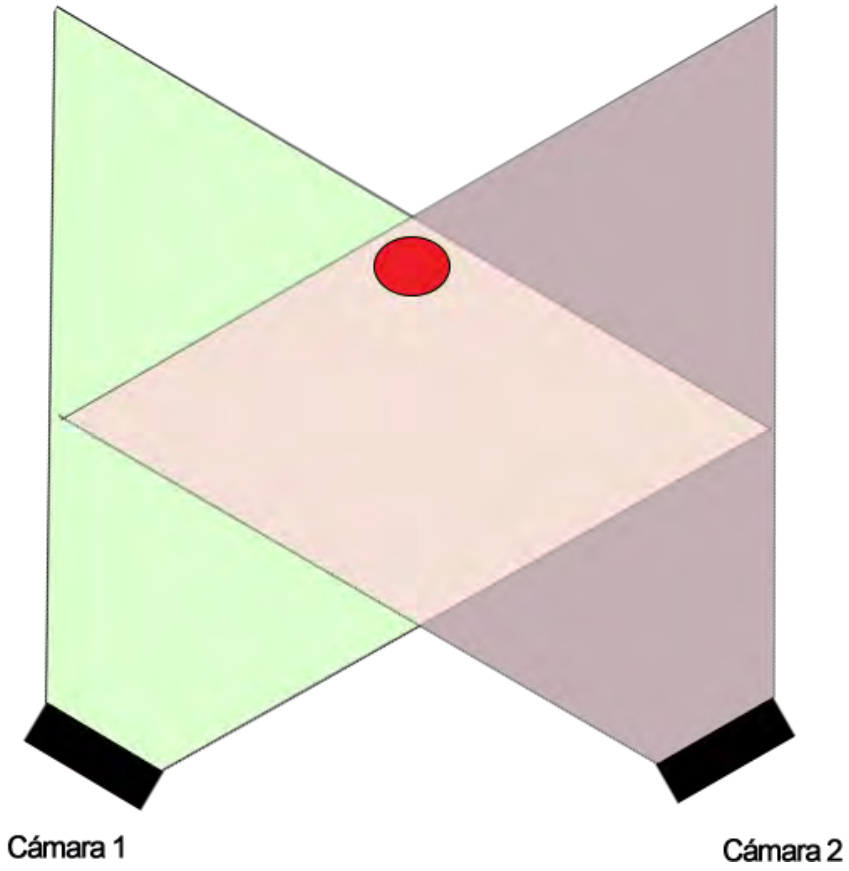

Figura 22 Posición de cámaras para este método. El punto rojo representa la posición deseada que debería tener el sujeto a la hora de capturar las imágenes.

Una vez que el sistema de cámaras está en la posición deseada, se capturan imágenes desde diferentes puntos de vista, en este caso el patrón se debe alcanzar a ver por completo por ambas cámaras (como se ve en la Figura 23) o se rechaza el par de imágenes en el cual no se alcance detectar el patrón en cualquiera de las dos imágenes.

Figura 23 Patrón de calibración visto desde dos cámaras distintas.

Para realizar la calibración se utiliza la aplicación "Stereo Camera Calibrator" que viene incluida en MATLAB. En esta se seleccionan los folders donde se encuentran las imágenes de cada cámara, que corresponderán como cámara 1 y cámara 2. Esto se hace por separado en distintos proyectos para las imágenes RGB e IR. Una vez que el mismo programa hace la detección del patrón se procede a calibrar y obtener los parámetros resultantes.

Se obtiene un objeto el cual contiene los parámetros de cada una de las cámaras (intrínsecos y extrínsecos). También incluye la matriz de rotación y el vector de traslación relativos de la cámara 2 a la cámara 1.

La matriz de rotación y vector de traslación relativos que se emplean son los obtenidos por las imágenes de IR, ya que al generar la nube de puntos se hace con respecto a los parámetros de IR y no de las de RGB. Teniendo los siguientes resultados:

Tabla 9 Posición relativa entre las cámaras.

Vector de translación	$(-1131.01424463337 \quad 38.5802121693769 \quad 505.242085834874)$
Matriz de rotación	$\begin{bmatrix} 0.576592828420242 & -0.0450425069897234 & -0.815789116609450 \\ 0.0484788258171572 & 0.998606109326143 & -0.0208720354515932 \\ 0.815592124570692 & -0.0275138325317204 & 0.577972728904817 \end{bmatrix}$

La persona se coloca aproximadamente en un punto medio entre las cámaras a una distancia de 1.5 [m] (Figura 24). Las imágenes se capturan simultáneamente por el software de captura para después obtener la nube de puntos. La transformada se aplica a la nube de puntos que fue capturada por la cámara que se definió como cámara 2.

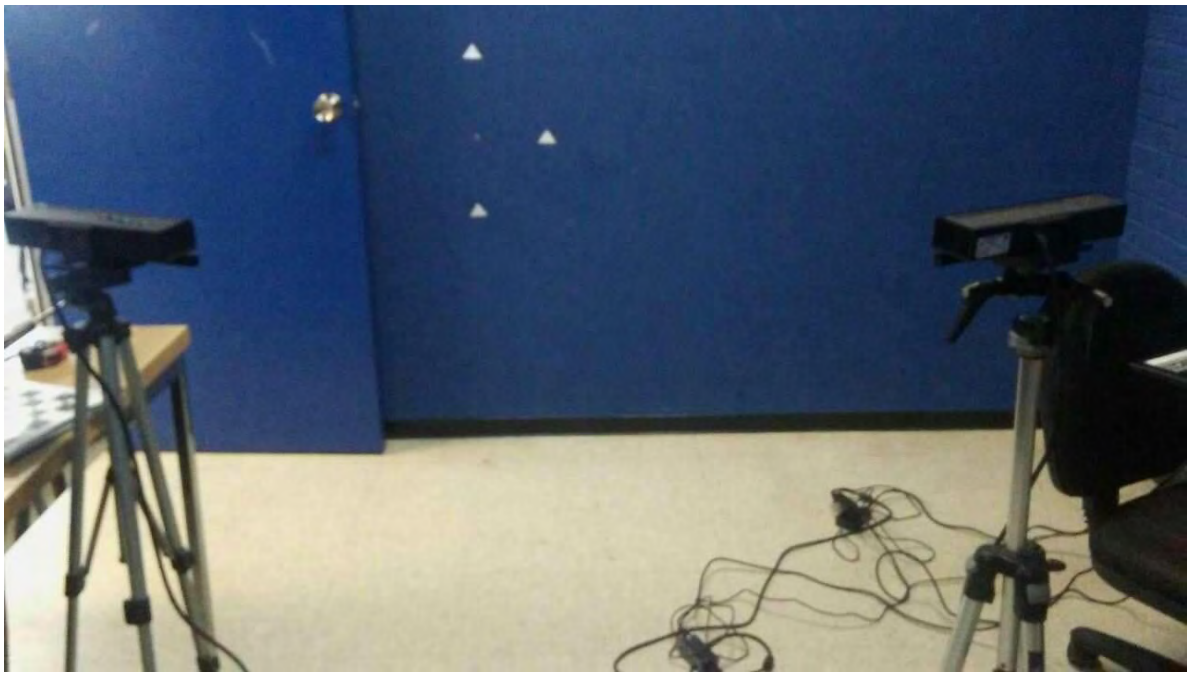

Figura 24 La persona se coloca entre las dos cámaras.

Se observa en la Figura 25 el resultado obtenido.

Figura 25 Ejemplo de nubes de puntos por medio de dos cámaras en posición lateral.

4 Obtención de puntos de referencia

En la obtención de los landmarks se emplean técnicas de procesamiento de imágenes basadas en operaciones de morfología discreta (erosión, dilatación, detección de últimos puntos) sobre las imágenes IR, debido a que por medio de los landmarks vistos en la cámara de infrarrojo se generan los puntos ciegos (Figura 26). El procedimiento consiste en obtener las coordenadas (X, Y, Z) de cada uno de los puntos y transformarlas a coordenadas del mundo de la cámara IR.

Figura 26 Los landmarks están colocados donde sean visibles para la cámara sobre un objeto, caso contrario no podrán ser detectados, estos se ven en la imagen como puntos ciegos sobre la imagen IR.

Al momento de capturar las imágenes se debe tener cuidado que no se encuentre algún otro tipo de objeto que pueda producir un reflejo (perillas, sillas de metal, etc.), ya que se puede confundir como un punto de referencia, algunos puntos generados por este error se filtran pero dependiendo del tamaño del objeto puede que no se filtre y produzca un error al contarlo como un punto de referencia de más.

En el caso que se encuentre menos puntos de referencia de una vista a la siguiente (principalmente cuando se hace uso de una sola cámara, por ejemplo, en el caso de la Figura 27), por medio de MATLAB se abre la vista que tenga más puntos y se hace selección del área que cubra los puntos que correspondan a la vista con menor puntos de referencia para crear una máscara en la imagen. Esta mascara hace que solo se contemple el área seleccionada eliminando el resto de la imagen (Figura 28), si la cantidad de puntos de referencia no es equivalente en ambas vistas, se da por hecho como error en las imágenes y da terminación al programa.

Figura 27 Ejemplo al hacer uso de una sola cámara, entre las vistas no se detectan la misma cantidad de landmarks.

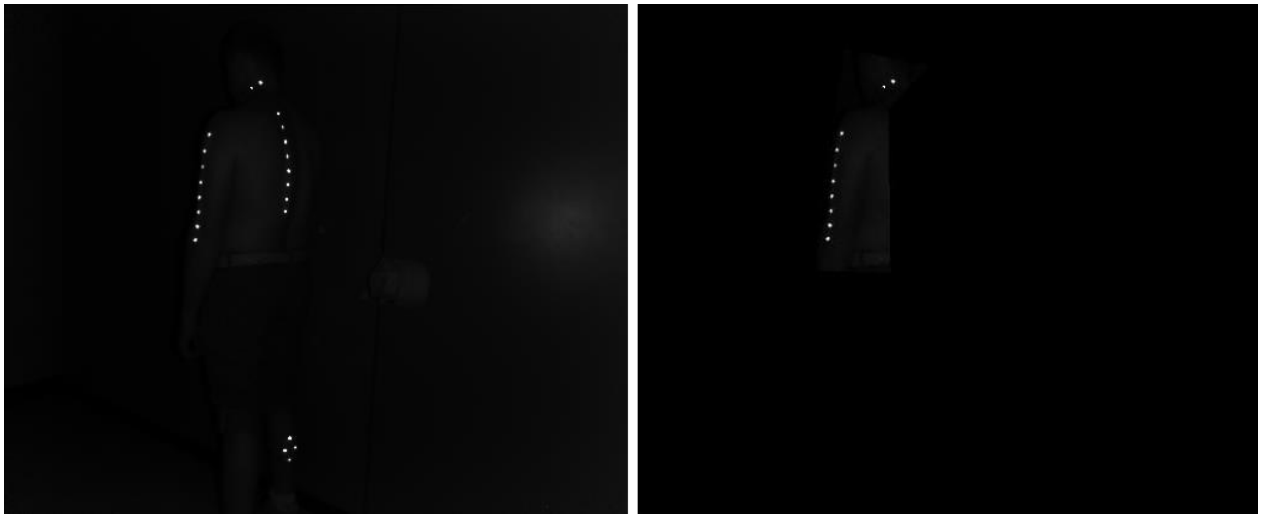

Figura 28 Ejemplo de máscara (imagen de la derecha) obtenida a partir de la imagen de la izquierda para obtener los landmarks que corresponden a la vista siguiente (la imagen de la izquierda de la Figura 27).

Al aplicar las operaciones morfológicas se obtienen los centroides de los puntos de referencia que corresponden a las coordenadas (X, Y), el mismo centroide se utiliza en la imagen de profundidad para obtener el valor de Z. Ya que puede existir un error mínimo en la obtención del centroide, este valor corresponde al promedio de pixeles de un área que rodea el punto.

Una vez obtenidas las coordenadas (X, Y, Z) de todos los puntos obtenidos se transforman por medio de las ecuaciones (1), (2), (3) respectivamente.

5 Registro rígido

El registro de imágenes es el método que permite el alineamiento de dos o más imágenes (tomando en cuenta que una imagen es el destino de otra) a causa de, por ejemplo, ser tomadas en distintas vistas, diferentes tiempos o diferentes sensores. Existen diversos métodos de registro como se detallan en [18] pero la idea principal de este procedimiento es determinar los puntos correspondientes entre las imágenes para luego determinar la transformada que existe entre ellas para obtener una imagen combinada entre ambas (Figura 29).

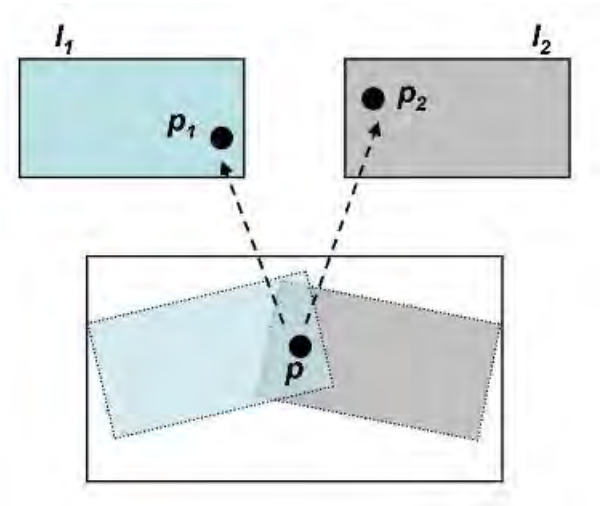

Figura 29 Un mismo punto visto en diferentes imágenes, al aplicar la transformada se obtiene una imagen combinada. Imagen tomada de [19].

El algoritmo implementado en este trabajo es el "punto más cercano iterativo" (*Iterative Closest Point*) o por sus siglas en ingles ICP, introducido en [20]. Es uno de los acercamientos al registro de figuras 3D para reducir la diferencia que hay entre puntos correspondientes. El registro empleado en [20] es el registro rígido (para objetos rígidos o bien figuras fijas), que se compone de una rotación R y una traslación t.

El método propuesto en [20] es el de la matriz de covarianza cruzada. Este método se explica a continuación:

Sabiendo que un punto de P corresponde a un punto de X, entonces

$$N_p = N_x \tag{10}$$

Donde

N_p es el numero de puntos del conjunto de puntos P.

N_x es el numero de puntos del conjunto de puntos X.

El vector de registro completo $\vec{q} = [\vec{q}_R|\vec{q}_T]^t$ es la transformación rígida óptima para el conjunto de puntos P hacia conjunto de puntos X.

Donde

$\vec{q}_T$ es el vector de traslación $\vec{q}_T = [q_X q_Y q_Z]^t$

$\vec{q}_R$ es el cuaternión unitario $\vec{q}_R = [q_w q_x q_y q_z]^t$ que genera la matriz de rotación de 3x3

Que minimiza la función objetivo cuadrática

$$f(\vec{q}) = \frac{1}{N_p}\sum_{i=1}^{Np} \|\vec{x}_i - R(\vec{q}_R)\vec{p}_i - \vec{q}_T\|^2 \tag{11}$$

Donde

$\vec{p}_i$ es el conjunto de puntos P que va a ser alineado con X

$\vec{x}_i$ es el conjunto de puntos X

$R(\vec{q}_R)$ es la matriz de rotación generada por el cuaternión unitario

En MATLAB ya se encuentra programado el método para transformar el cuaternión unitario $R(\vec{q}_R)$ a la matriz 3x3 de rotación.

Para calcular la matriz de covarianza cruzada $\sum_{px}$ se necesita encontrar el centro de masa de P y X por medio de las siguientes ecuaciones:

$$\vec{\mu}_p = \frac{1}{N_p}\sum_{i=1}^{Np} \vec{p}_i \tag{12}$$

$$\vec{\mu}_x = \frac{1}{N_x}\sum_{i=1}^{Nx} \vec{p}_x \tag{13}$$

A continuación se obtiene la matriz de covarianza cruzada que se calcula por medio de la siguiente ecuación

$$\sum_{px} = \frac{1}{N_p}\sum_{i=1}^{Np} [\vec{p}_i \vec{x}_i^t] - \vec{\mu}_p \vec{\mu}_x^t \tag{14}$$

Los componentes cíclicos de la matriz anti-simétrica $A = (\sum_{px} - \sum_{px}^T)$ son usados para formar el vector $\Delta = [A_{23}\ A_{31}\ A_{12}]^T$. El vector obtenido se utiliza para crear la matriz simétrica $Q(\sum_{px})$ que está dada por

$$Q(\sum_{px}) = \begin{bmatrix} tr(\sum_{px}) & \Delta^T \\ \Delta & \sum_{px} + \sum_{px}{}^T - tr(\sum_{px}) I_3 \end{bmatrix} \tag{15}$$

Donde

I_3 es la matriz de identidad 3x3

El vector propio $\vec{q}_R = (q_0\ q_1\ q_2\ q_3)^T$ corresponde al máximo valor propio de la matriz $Q(\sum_{px})$ que es seleccionado como la rotación óptima. Este vector, como se mencionó previamente, se transforma a la matriz de rotación.

El vector de traslación óptimo está dado por

$$\vec{q}_T = \vec{\mu}_x - R(\vec{q}_R)\vec{\mu}_p \tag{16}$$

Esta operación, cuaternión de mínimos cuadrados, se denomina como

$$(\vec{q}, d_{ms}) = Q(P, X) \tag{17}$$

Donde

$\vec{q}$ es la transformación rígida óptima

d_{ms} es el error cuadrático de los puntos de coincidencia.

5.1 ICP

El algoritmo ICP propuesto en [20] es el más común entre sus variantes y el cual se implementa en este trabajo. A continuación se explica cómo funciona:

Dados un conjunto de puntos P y un conjunto de puntos X, donde se sabe que $N_p = N_x$, se busca minimizar la distancia cuadrática entre ellos por medio del cálculo de la distancia euclidiana

$$d(\vec{p}, X) = \min_{\vec{x} \in X} \|\vec{x} - \vec{p}\| \tag{18}$$

para encontrar su punto más cercano de cada punto.

Una vez encontrado el punto más cercano de cada uno de los puntos de P a X, se obtiene el registro de mínimos cuadrados $\vec{q}$ como se describió previamente. Por último, se actualiza el conjunto de puntos P por medio de la transformada obtenida (Figura 30).

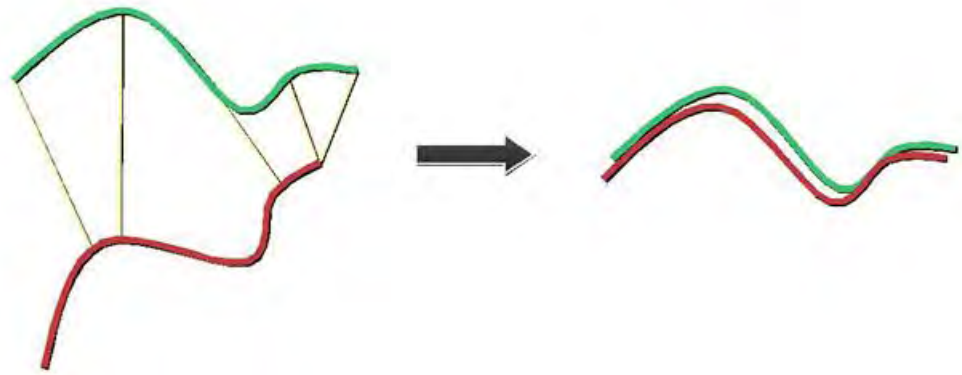

Figura 30 El algoritmo ICP intenta minimizar la distancia entre los puntos y lograr hacer una alineación lo mejor aproximada posible. Imagen tomada de [21].

Los pasos del algoritmo son los siguientes

1. Se calculan los puntos más cercanos $Y_k = C(P_k, X)$
2. Se realiza el registro $(\vec{q}_k, d_k) = Q(P_0, Y_k)$
3. Se aplica la transformada a los puntos $P_{k+1} = \vec{q}_k(P_0)$
4. Se calcula el error $d_k - d_{k+1}$ y se repite el procedimiento hasta que el error sea menor a la tolerancia definida

6 Experimentación y Resultados

En este trabajo se realizaron pruebas de 3 formas diferentes para escanear al sujeto: con una cámara y con dos cámaras (en posición frontal y lateral). Previamente se realizó la calibración de las cámaras y se encontró la posición relativa de la segunda cámara para las pruebas que hacen uso de más de un Kinect.

El proceso para todos los métodos consiste en los siguientes pasos:

- Colocación de puntos de referencia en el cuerpo del sujeto.
- Captura de imágenes con el Kinect por medio del software de captura.
- Procesamiento de imágenes para la obtención de puntos de referencia.
- Generación de nube de puntos de cada vista.
- Alineación de las nubes de puntos mediante el algoritmo ICP (*Iterative Closest Point*) aplicado a los puntos de referencia.
- Reconstrucción de las mallas 3D mediante software de modelado.

Para la reconstrucción de la malla en cada uno de los casos se empleó el software Meshlab una vez obtenida la nube de puntos final. En este trabajo no se abarca el método de reconstrucción de mallas al solo emplear el tutorial descrito en [22] el cual emplea el método de Reconstrucción de Superficie de Poisson (*Poisson Surface Reconstruction*).

6.1 Pruebas con una cámara

Para las pruebas realizadas con una cámara se utiliza el procedimiento de calibración previamente explicado en el capítulo 3.3.1.

La cámara se coloca en 5 posiciones diferentes, capturando las imágenes por cada posición, a una distancia aproximadamente de 1.5 [m] alrededor de una circunferencia, como se ve en la Figura 31.

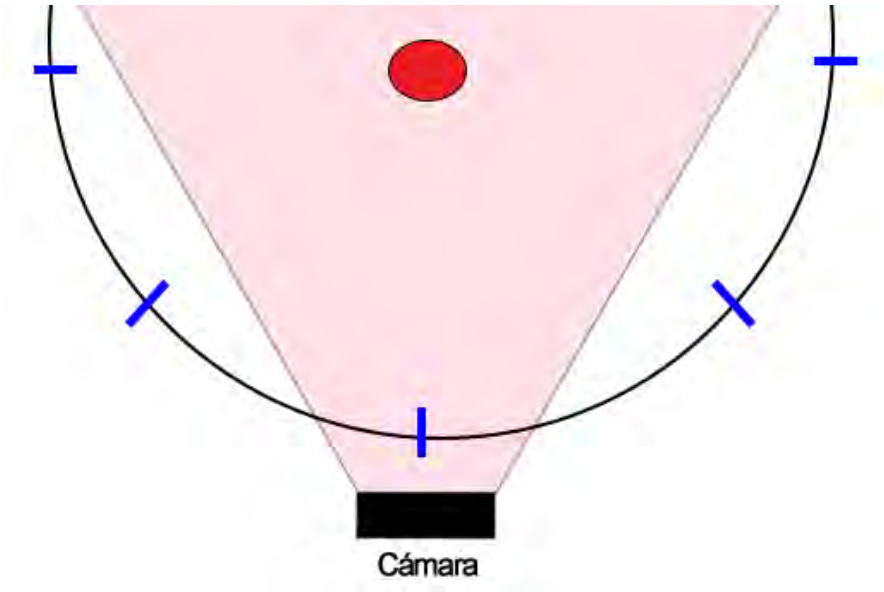

Figura 31 La cámara se coloca en 6 posiciones distintas para capturar al sujeto, las posiciones deseadas se encuentran marcadas por líneas azules. El punto rojo representa la posición que debe tomar la persona.

Durante este experimento se realizaron pruebas previas con un objeto rígido para probar los algoritmos. En este caso se utilizó una caja colocando solamente 8 landmarks sobre está. En la Figura 32 se puede ver un ejemplo de los puntos de referencia sin haber aplicado la transformada y en la Figura 33 los puntos de referencia una vez al haberle aplicado la transformada.

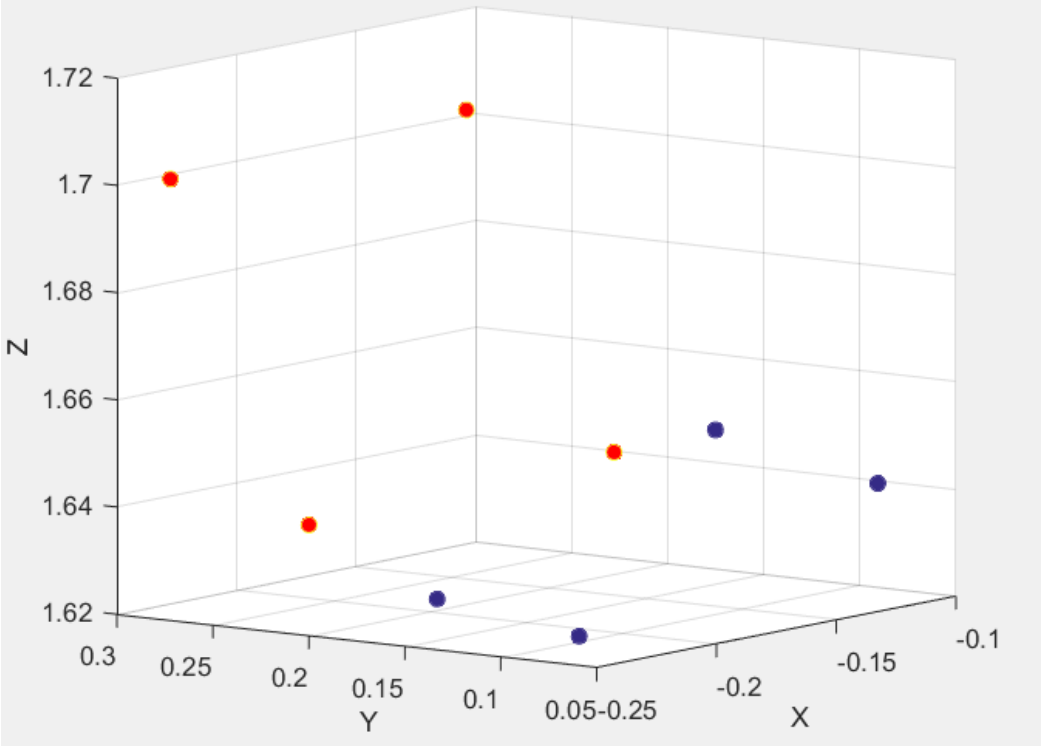

Figura 32 Landmarks de la caja en su posición inicial capturada. Los puntos azules se van alinear con los puntos rojos.

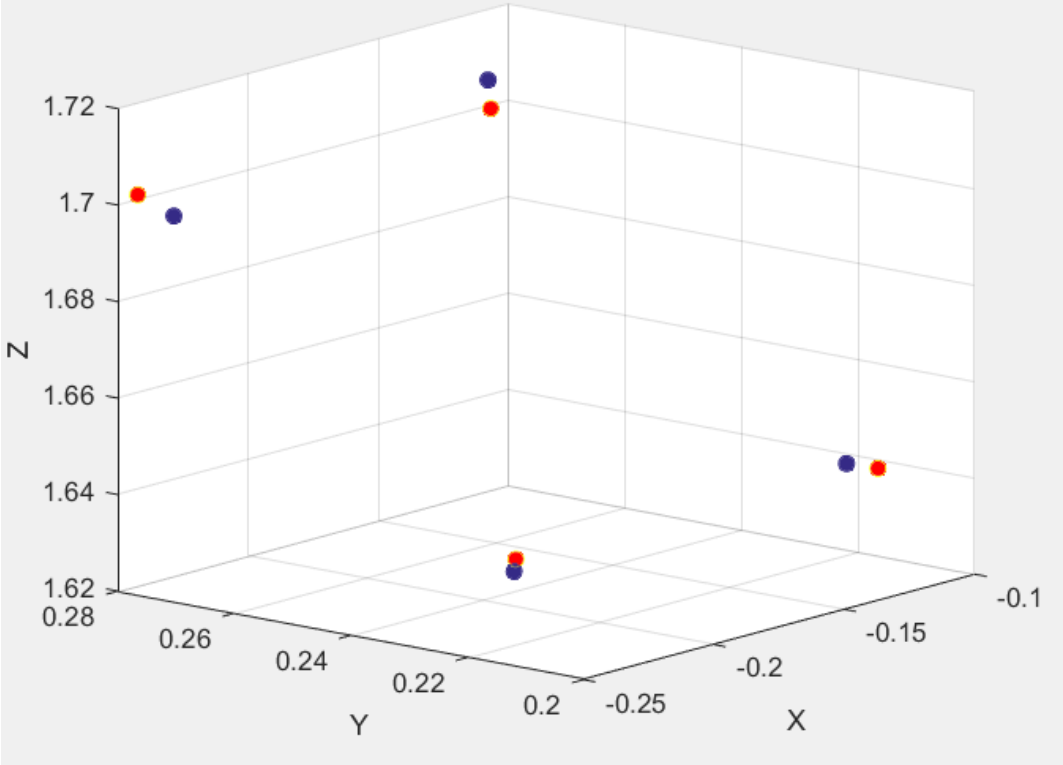

Figura 33 Landmarks de la caja una vez que se le aplico la transformación óptima.

El resultado de la nube de puntos se puede ver en la Figura 34.

Figura 34 Nube de puntos de una caja sobre una mesa, se alinearon 6 vistas. En este caso no se segmenta el objeto.

Al comprobar el funcionamiento del algoritmo ICP, se realizó el experimento con una persona. En este se hicieron diferentes pruebas con un número variable de landmarks. La mejor respuesta en el registro de los puntos fue al colocarle 38 landmarks alrededor del cuerpo (pecho, espalda, brazos y piernas). En la Figura 35 se pueden observar todas las imágenes IR que se obtuvieron.

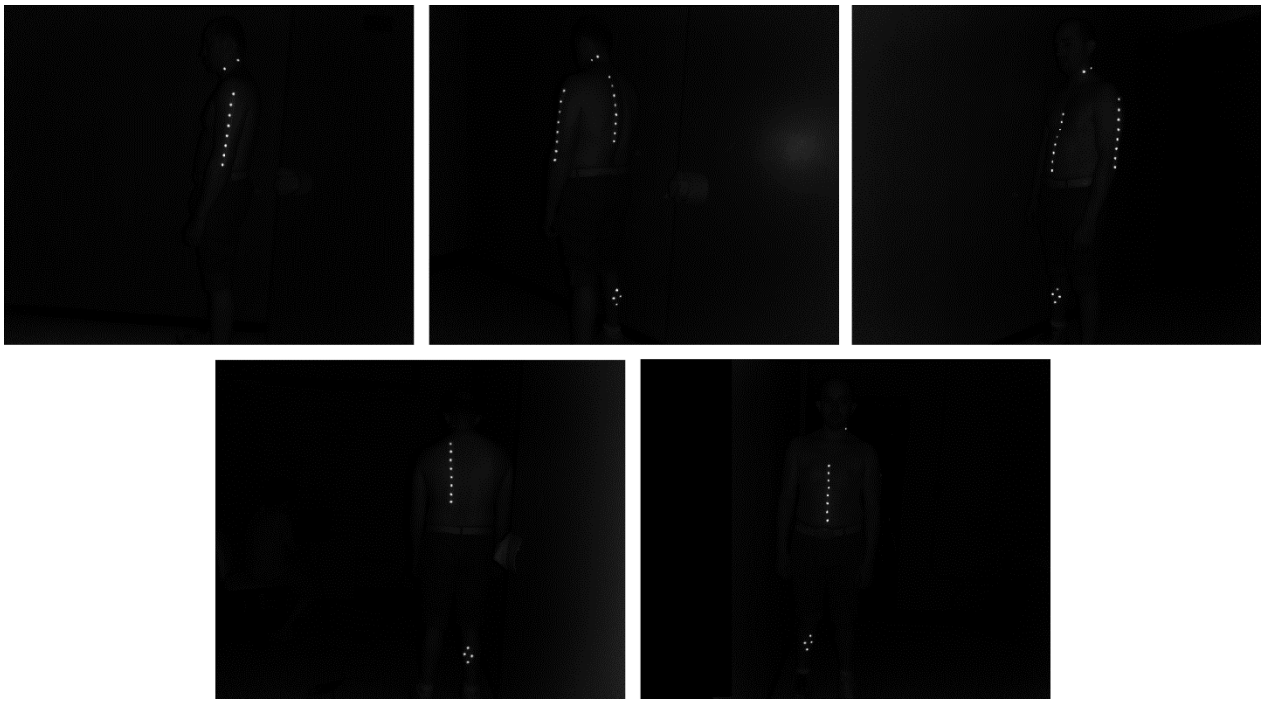

Figura 35 Imágenes IR utilizadas en la prueba para obtener los landmarks sobre el cuerpo del sujeto (se le colocaron en total 38 landmarks alrededor del cuerpo).

Una vez iniciada la captura de imágenes, la persona intenta quedarse en la misma posición durante la prueba (debido a que esta puede que se mueva) y el Kinect se va colocando en diferentes posiciones indicadas en el suelo previamente, capturando las imágenes por cada vista. En la Figura 36 se pueden observar las imágenes RGB obtenidas para esta prueba.

Figura 36 Imágenes RGB utilizadas en la prueba para el mapeo de las nubes de puntos.

El resultado de las 6 nubes de puntos se puede ver en la Figura 37 y la nube de punto final se ve en la Figura 38.

Figura 37 Nubes de puntos obtenidas en esta prueba para este método.

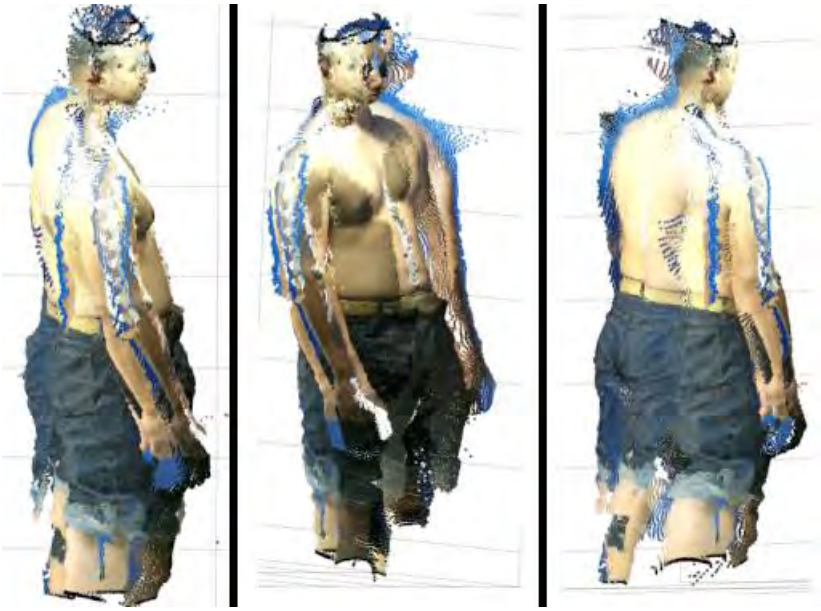

Figura 38 Nube de puntos final, después de haberles aplicado la transformada optima obtenida a las nubes de puntos.

El modelo reconstruido con malla se puede observar en la Figura 39.

Figura 39 Modelo con malla reconstruida. La cara al tener muchos vértices no alineados se llega a desfigurar, al igual que varias regiones del cuerpo.

6.2 Pruebas con cámaras frontales

En este caso, previamente se calibran y se colocan las cámaras como se explica en el capítulo 3.4.

En las pruebas de este experimento desarrollado se hace la captura de imágenes de 3 vistas por cámara (frontal y trasera) dando un total de 6 vistas, a causa de que por cada vista se hace la unión entre la nube de puntos tomada por la cámara frontal y la cámara trasera como se explica en el capítulo 3.4. Por lo tanto en las pruebas desarrolladas solo se necesitó hacer el registro de 3 nubes de puntos.

La persona se coloca entre ambas cámaras dirigiéndose a una de ellas sin importar a cual, mientras esté en dirección hacia ésta (esta será la posición inicial). Para la segunda vista el sujeto gira alrededor de su eje alrededor de 30 grados la izquierda y por ultimo luego a la derecha alrededor de 30 grados a partir de su posición inicial. Entre cada vista se queda quieta la persona hasta que termine la grabación de las imágenes. Un ejemplo de las posiciones que se coloca la persona se ve en la Figura 40.

Figura 40 Ejemplo de vistas capturadas, de izquierda a derecha las posiciones son: frontal, giro a la derecha y giro a la izquierda. Las imágenes superiores son las imágenes frontales y las inferiores son las traseras.

En este caso las pruebas se hicieron solamente con 8 landmarks colocados en la parte frontal de los brazos de la persona como se ve en la Figura 41.

Figura 41 Imagen IR. Los 8 landmarks se colocaron solamente en la parte frontal de los brazos.

En la Figura 42 se ve un ejemplo de cada nube de puntos (formada por su parte frontal y trasera) por separado y el resultado al realizar el registro de las nubes de puntos se ve en la Figura 43.

Figura 42 Ejemplos de nubes de puntos de prueba realizada con el segundo método. De izquierda a derecha las nubes de puntos son: frontal, giro a la derecha y giro a la derecha.

Figura 43 Ejemplo de nubes de puntos alineadas en el segundo caso.

En la Figura 43 se ve los modelos 3D de las pruebas realizadas.

Figura 44 Ejemplos de modelos con malla reconstruida para el segundo caso.

6.3 Pruebas con cámaras laterales

En este caso las cámaras se posicionan y calibran como se explica en el capítulo 3.5.

A la persona se le colocan 8 landmarks en la parte frontal de la persona. En este caso se toman 3 vistas por cámara, teniendo un total de 6 las cuales se combinan para tener un total de 3 nubes de puntos.

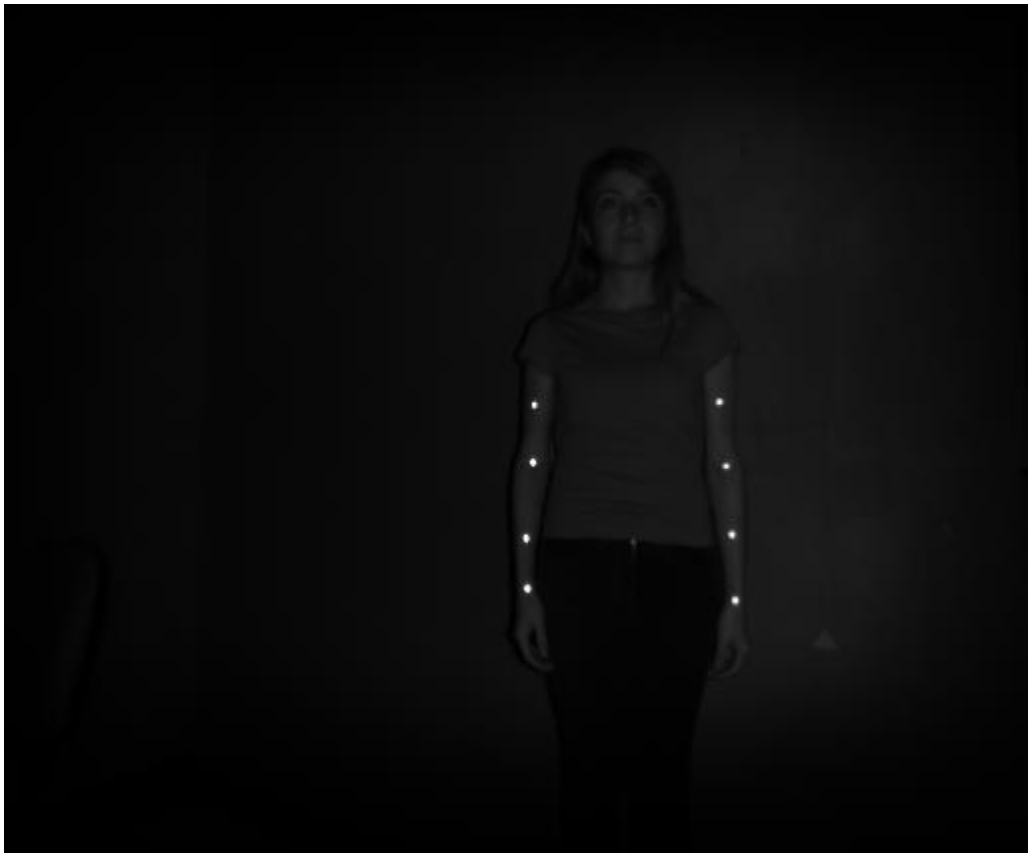

Figura 45 Imagen IR. Los 8 landmarks se colocaron sobre de los brazos.

También que en el caso anterior, la persona se debe mantener fija al momento de capturar las imágenes y girar en su eje, las posiciones son: Frontal (que se toma como la posición inicial), girar alrededor de 20 grados a la derecha y luego a la izquierda a partir de su posición inicial. Ejemplo de las capturas se observa en la Figura 46.

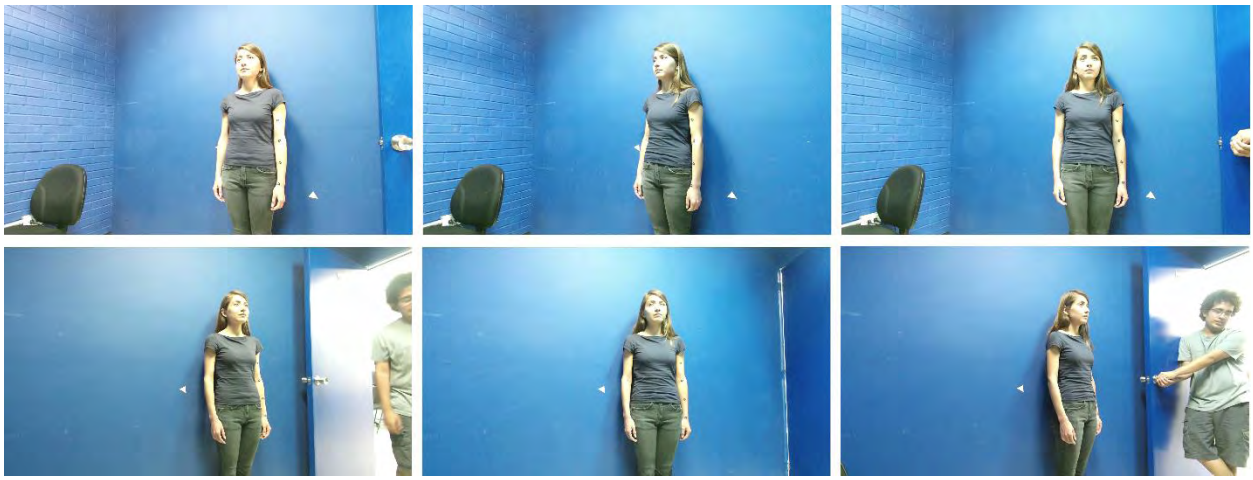

Figura 46 Prueba para el tercer método de captura. Las imágenes superiores son capturadas por una cámara, las imágenes inferiores son capturadas por otra cámara.

Figura 47 Nubes de puntos generadas durante la prueba.

Una vez obtenidas todas las nubes de puntos (Figura 47), se emplea solamente las imágenes IR de una sola cámara para hacer el registro entre las nubes de puntos por medio del procedimiento previamente explicado.

El resultado de la nube de puntos Figura 48.

Figura 48 Nubes de puntos alineadas para la prueba realizada en el tercer caso.

El modelo con malla se puede ver en la Figura 49

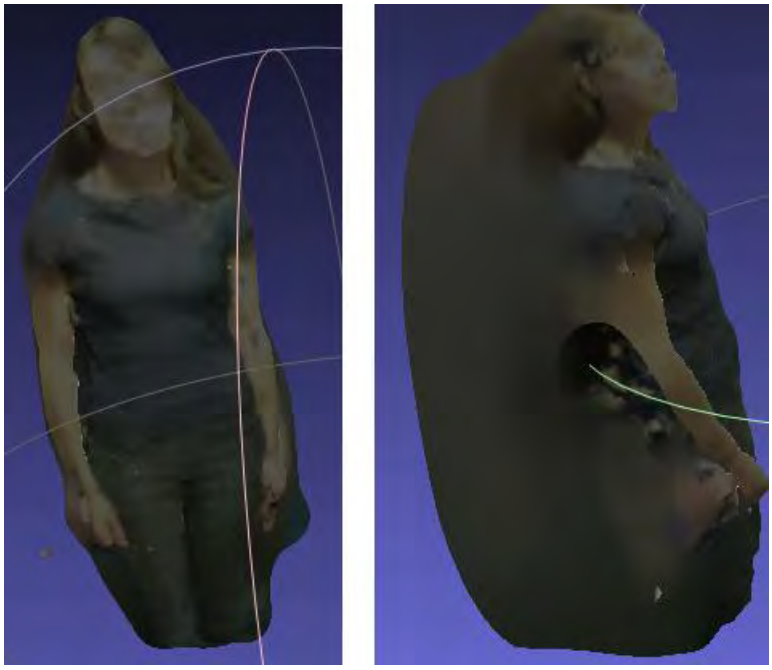

Figura 49 Ejemplo de modelo con malla reconstruida. En este caso al solo tener la parte frontal no se cierra la malla por el método de reconstrucción de malla utilizado.

7 Conclusiones

En este trabajo se realizaron pruebas por medio de 3 métodos distintos para escanear al sujeto con resultados variantes.

En el primer caso (con una sola cámara) se logró obtener un modelo poco detallado, la alienación no llegó a ser la deseada debido a que algunas regiones del cuerpo se alinearon mejor que otras. Además, se llegó a tener una mayor complejidad al escanear a los individuos en comparación a los otros métodos a causa de 3 razones principales que afectan los resultados: se requirió un mayor control sobre la cámara (las posiciones en las que se colocaba), la cantidad de puntos de referencia que se debe colocar en la persona es mucho mayor, y por último, es el experimento que tomaba más tiempo en capturar todas las vistas, siendo el más agotador para la persona, haciendo que a veces llegara a perder el equilibrio o se moviera al haber mantenido una misma posición por un largo periodo de tiempo.

En el segundo caso (con dos cámaras en posiciones frontales) el resultado fue un modelo lo suficientemente detallado, siendo un mejor resultado al esperado y mejor a comparación de todas las pruebas realizadas en este trabajo, igualmente cuenta con una cantidad mucho menor de puntos de referencia, llego a ser mucho más rápido de realizar y por ende la persona no debe mantener por mucho tiempo una posición. Uno de los problemas principales fue el movimiento que necesita hacer la misma persona (al tener fijas las cámaras) en este método. En las pruebas realizadas al momento de girar las personas se colocaban en una posición poco diferente, realizaban movimientos con la cabeza o brazos alterando el modelo final.

Para el tercer y último caso (empleando dos cámaras de manera lateral) el resultado logró una alineación decente al esperarse un mejor resultado, el problema se presentó al posicionar las cámaras, a causa de que en la prueba realizada se colocó una cámara más adelante que la otra (siendo un error humano), provocando que, aun alineadas, no hubiera una alineación completa al tener la nube de puntos de la persona a diferentes tamaños.

El empleo del Kinect v2 demostró ser capaz, al ser una cámara de bajo costo que permite realizar la reconstrucción 3D, poder generar nubes de puntos lo suficientemente detalladas para que un individuo se pueda identificar con el mismo.

Asimismo, los puntos de referencia utilizados en este trabajo logro demostrar ser una opción para realizar el registro entre las nubes de puntos, no obstante, el poner grandes cantidades de puntos de referencia puede llegar a ser tardado prolongando el escaneo del sujeto, así como también a veces se presentaron problemas (debido al error humano) al colocarlos en posiciones donde no estén a la vista de la o las cámaras a la hora de capturar las imágenes. También el problema con estos fue al tener que tener cuidado con objetos que produzcan un brillo necesitando tener un ambiente más controlado. En general el uso de los puntos de referencia resulto ser posible y una opción para la alineación de las nubes de puntos.

Los problemas principales en este trabajo fueron al mover la cámara y al momento que la persona tiene que colocarse en distintas posiciones (como ya se comentó previamente en el caso 2 y 3) que modifican el modelo resultante aunque el registro rígido obtenga la transformación óptima por medio del algoritmo ICP.

7.1 Trabajo a futuro

El programa actual de este trabajo cuanta solamente con los algoritmos necesarios para ser ejecutados sobre la plataforma de MATLAB por lo que se requeriría construir una interfaz con Qt o Visual Studio que llegue a ser más amigable para los usuarios que no conocen por completo el programa.

Actualmente el software solamente captura alrededor de un 80%-90% del sujeto, principalmente faltando los pies, al no contar con un motor como el Kinect v1, una posible solución es encontrar una distancia estándar y una altura para el Kinect v2 alcance a detectar todo el cuerpo y los puntos de referencia.

Para el error humano de colocar el o los Kinects a diferentes distancias o por el movimiento causado por la persona (agotamiento, desequilibrio, etc.) debido a que debe estar en una posición estática por un tiempo considerable se piensa adecuado hacer el escalamiento entre las imágenes o implementar un tipo de registro a fin que permita hacer el escalamiento (registro no-rígido) entre ellas.

La implementación de un algoritmo que elimine los puntos no confiables de la nube de puntos del cuerpo humano, es decir, aun mediante la segmentación y filtrando la mayor parte de los puntos que no pertenecen al cuerpo, existen aún puntos que producen ruido sobre las nubes de puntos y la nube de puntos final, afectando al momento de generar la malla del modelo.

También sería necesario probar software o métodos alternativos para la generación de la malla, los modelos actuales pierden mucho detalle en el cuerpo y principalmente el rostro por el método usado de Meshlab por lo que se debe probar alternativas para hacer una comparación si hay resultados diferentes.

8 Apéndice

8.1 Demostración de matriz P_{rgb}

P_{rgb} es la matriz de transformación que hace la relación relativa entre las cámaras RGB e IR. Nos sirve para pasar de las coordenadas de C_{ir} a C_{rgb}.

La siguiente demostración se puede encontrar en [23].

Dadas las matrices de proyección de las cámaras RGB e IR

$$P_{ir}^{\{W\}} = \begin{bmatrix} R_{ir} & t_{ir} \\ 0 & 1 \end{bmatrix} \text{ y } P_{rgb}^{\{W\}} = \begin{bmatrix} R_{rgb} & t_{rgb} \\ 0 & 1 \end{bmatrix}$$

Tomando en cuenta que capturan el mismo cuadro y definiendo a W como el cuadro global (world frame). A partir de C_{ir} se transforman las coordenadas a W para después a transformarlas a coordenadas de C_{rgb}.

Dado un punto $q^{\{ir\}}$ en C_{ir}, la transformada a coordenadas W se define por

$$q^{\{W\}} = t_{ir} + R_{ir} q^{\{ir\}} \ldots [1]$$

Partiendo de la transformada de C_{rgb} a W se despeja $q^{\{rgb\}}$

$$q^{\{W\}} = t_{rgb} + R_{rgb} q^{\{rgb\}}$$

$$q^{\{W\}} - t_{rgb} = R_{rgb} q^{\{rgb\}}$$

$$R_{rgb}^{-1}(q^{\{W\}} - t_{rgb}) = q^{\{rgb\}}$$

Por lo tanto dado un punto $q^{\{W\}}$ en W, las coordenadas C_{rgb} están dadas por

$$q^{\{rgb\}} = R_{rgb}^{-1}(q^{\{W\}} - t_{rgb}) \ldots [2]$$

Sustituyendo la ecuación [1] en [2]

$$q^{\{rgb\}} = R_{rgb}^{-1}((t_{ir} + R_{ir} q^{\{ir\}}) - t_{rgb})$$

$$q^{\{rgb\}} = R_{rgb}^{-1}(R_{ir} q^{\{ir\}} + t_{ir} - t_{rgb})$$

$$q^{\{rgb\}} = R_{rgb}^{-1} R_{ir} q^{\{ir\}} + R_{rgb}^{-1}(t_{ir} - t_{rgb})$$

Que se define como la matriz P_{rgb}

$$P_{rgb} = P_{ir}^{\{rgb\}} = \begin{bmatrix} R_{rgb}^{-1} R_{ir} & R_{rgb}^{-1}\left(t_{ir} - t_{rgb}\right) \\ 0 & 1 \end{bmatrix}$$

Por lo tanto las coordenadas de IR a RGB se pueden obtener como

$$q^{\{rgb\}} = P_{rgb} q^{\{ir\}}$$

9 Bibliografía

[1] P. Palasek, Heng Yang, Zongyi Xu, N. Hajimirza, E. Izquierdo, and I. Patras, “A flexible calibration method of multiple Kinects for 3D human reconstruction,” in *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 2015, pp. 1–4.

[2] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan, “Scanning 3D Full Human Bodies Using Kinects,” *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 4, pp. 643–650, Apr. 2012.

[3] R. Wang, M. Hernandez, J. Choi, and G. Medioni, *Accurate 3D Face and Body Modeling from a Single Fixed Kinect*. 2013.

[4] A. Mao, H. Zhang, Y. Liu, Y. Zheng, G. Li, and G. Han, “Easy and Fast Reconstruction of a 3D Avatar with an RGB-D Sensor,” *Sensors*, vol. 17, no. 5, 2017.

[5] Y. Cui, W. Chang, T. Nöll, and D. Stricker, “KinectAvatar: Fully Automatic Body Capture Using a Single Kinect,” in *Computer Vision - ACCV 2012 Workshops*, 2013, pp. 133–147.

[6] H. Pang, J. Li, J. Peng, X. Zhong, and X. Cai, “Personalized full-body reconstruction based on single kinect,” in *2015 8th International Congress on Image and Signal Processing (CISP)*, 2015, pp. 979–983.

[7] G. Zhang, J. Li, J. Peng, H. Pang, and X. Jiao, “3D human body modeling based on single Kinect,” in *2014 7th International Conference on Biomedical Engineering and Informatics*, 2014, pp. 100–104.

[8] Y. Chen and Z.-Q. Cheng, *Personalized avatar capture using two Kinects in a moment*. 2012.

[9] Z. Liu *et al.*, *3D real human reconstruction via multiple low-cost depth cameras*, vol. 112. 2015.

[10] S. R. Deepika and N. Avinash, “Real-Time Automatic Detection and Recognition of Hamming Code Based Fiducial Marker,” in *Proceedings of the Fourth International Conference on Signal and Image Processing 2012 (ICSIP 2012)*, 2013, pp. 575–584.

[11] A. Chatterjee, *Geometric Calibration and Shape Refinement for 3D Reconstruction*. 2015.

[12] “Radial Distortion Correction.” [Online]. Available: http://www.uni-koeln.de/~al001/radcor_files/hs100.htm. [Accessed: 18-May-2018].

[13] Z. Zhang, “A Flexible New Technique for Camera Calibration,” *IEEE Trans Pattern Anal Mach Intell*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[14] E. Lachat, H. Macher, M-A Mittet, T. Landes, and P. Grussenmeyer, “FIRST EXPERIENCES WITH KINECT V2 SENSOR FOR CLOSE RANGE 3D MODELLING,” in *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2015, vol. XL, pp. 93–100.

[15] “Nicolas Burrus Homepage - Kinect Calibration.” [Online]. Available: http://burrus.name/index.php/Research/KinectCalibration. [Accessed: 10-Mar-2018].

[16] L. Almeida, F. Vasconcelos, J. P. Barreto, P. Menezes, and J. Dias, *On-line incremental 3D human body reconstruction for HMI or AR applications*. 2011.

[17] “Kinect SDK C++ Tutorials - 3. Point Clouds.” [Online]. Available: https://homes.cs.washington.edu/~edzhang/tutorials/kinect2/kinect3.html. [Accessed: 21-May-2018].

[18] B. Zitová and J. Flusser, “Image registration methods: a survey,” *Image Vis. Comput.*, vol. 21, no. 11, pp. 977–1000, Oct. 2003.

[19] L. W. Kheng, “Image Registration,” p. 39.

[20] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[21] “3D full object reconstruction from Kinect – Signal and Image Processing Lab.” .

[22] M. Tenney, “Point Clouds to Mesh in ‘MeshLab,’” *Geospatial Modeling & Visualization*, 04-Jun-2012. [Online]. Available: http://gmv.cast.uark.edu/scanning/point-clouds-to-mesh-in-meshlab/. [Accessed: 28-May-2018].

[23] “geometry - Relative camera matrix (pose) from global camera matrixes - Mathematics Stack Exchange.” [Online]. Available: https://math.stackexchange.com/questions/709622/relative-camera-matrix-pose-from-global-camera-matrixes. [Accessed: 12-Mar-2018].